

142622

KARA KUVVETLERİ MESAJ SİSTEMİNDE
XML TEKNOLOJİLERİNİN KULLANILMASI

142622-

YÜKSEK LİSANS TEZİ
Hayrettin DÖNMEZ
514011050

Tezin Enstitüye Verildiği Tarih : 24 Nisan 2003

Tezin Savunulduğu Tarih : 30 Mayıs 2003

Tez Danışmanı :

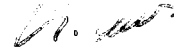
Doç.Dr. A.Coşkun SÖNMEZ

Diğer Jüri Üyeleri

Prof.Dr. Eşref ADALI

Doç.Dr. Serhat ŞEKER







MAYIS 2003



ÖNSÖZ

Bu çalışmanın hazırlanmasında bana yardımlarını esirgemeyen, yol gösteren değerli tez danışmanım Doç Dr. A. Coşkun Sönmez'e ve katkılarını asla unutamayacağım Doç.Dr. Serhat Şeker'e sonsuz teşekkürlerimi bir borç bilirim.

Bütün fakülte çalışanlarına bizlere gösterdikleri dostluk ve yardımlarından dolayı ayrıca teşekkür ederim.

Ayrıca her zaman yanımda olan ve destekleyen değerli aileme ve arkadaşlarıma buradan bir kez daha teşekkür etmenin mutluluğunu yaşıyorum.

Mayıs, 2003

Hayrettin DÖNMEZ

İÇİNDEKİLER

ÖNSÖZ.....	i
KISALTMALAR	iv
TABLO LİSTESİ.....	v
ŞEKİL LİSTESİ.....	vi
ÖZET	vii
SUMMARY	viii
1.GİRİŞ.....	1
2. XML NEDİR	2
2.1 Giriş.....	2
2.2 İşaretleme Dilleri Ve Tarihçesi	2
2.3 Neden XML	3
2.4 XML Tabanlı Teknolojiler Ve Uygulamaları.....	5
2.5 XML Söz Dizimi.....	7
2.5.1 Elemanlar	7
2.5.2 Özellikler.....	8
2.5.3 Karakter Veri	8
2.5.4 Karakter ve Varlık referansları	9
2.5.5 Açıklamalar.....	9
2.5.6 İşlem Komutları	9
2.5.7 CDATA bölümleri	9
3. XML VERİ TİPİ TANIMLAMALARI (DTD)	10
3.1 Belge Tipi Deklarasyonu	10
3.2 İyi Oluşmuş ve Doğru Belgeler	11
3.3 DTD Söz Dizimi(Syntax)	11
3.3.1 Elemanlar	12
3.3.2 Özellikler (Attributes).....	13
3.3.3 Varlıklar (Entities)	16
3.3.4 Notation (Gösterim).....	17
4. XML İLGİLİ TEKNOLOJİLER.....	18
4.1 Ayrıştırma (Xml Parsing).....	18
4.2 SAX (Simple API for XML).....	19
4.3 DOM (Document Object Model)	20
4.4 XML Parser'ları (XML Ayrıştırıcıları)	21
4.5 XPath.....	22
4.6 XPointer	24

4.7 XLink	25
4.8 XML Şema (XML Schema).....	25
4.9 XML Belgelerinin Sunumu	26
4.9.1 XSL (eXtensible Stylesheet Language)	26
4.9.2 CSS (Cascading Style Sheets)	28
4.10 XQuery.....	29
4.11 XHTML	30
5. BENZER VE İLGİLİ ÇALIŞMALAR.....	33
5.1. Scenario Authoring And Visualization for Advanced Graphical Environments (Savage).....	33
5.2. Visualization Of Theater-Level Radio Communications Using A Networked Virtual Environment.....	34
5.3. OASIS ebXML Message Service Protocol.....	35
5.4. XML Messaging Protocol (XMLMSG 1.0).....	35
5.5 Java™ API for XML Messaging (JAXM)	36
5.6. Simple Object Access Protocol (SOAP)	36
5.7.XML/EDI.....	37
6. UYGULAMA.....	38
6.1. Geliştirme Ortamı.....	38
6.2. Uygulama için oluşturulan XML dokümanları ve yapıları	39
6.3. Uygulama Modülleri.....	42
6.3.1. Giriş Ekranı Modülü	42
6.3.2. Mesaj Merkezi Modülü.....	43
6.3.2.1. Gelen Mesaj Kutusu Modülü	43
6.3.2.2. Giden Mesaj Kutusu Modülü.....	45
6.3.2.3. Mesaj Gönderme Modülü	45
6.3.2.4. Üyelik Bilgileri Modülü.....	47
6.3.2.5. Online Hat Modülü	47
6.3.2.6. Çıkış Modülü	48
6.3.3. Yeni Üye Başvuru Modülü	48
6.3.4. Sistem Bilgileri Modülü.....	48
7. SONUÇLAR VE ÖNERİLER	49
KAYNAKLAR	51
EKLER.....	53
ÖZGEÇMİŞ.....	54

KISALTMALAR

XML	: Extensible Markup Language (Genişleyebilir İşaretleme Dili)
GML	: Generalized Markup Language (Genel İşaretleme Dili)
ISO	: The International Organization for Standardization
SGML	: Standart Generalized Markup Language
HTML	: Hypertext Markup Language
UML	: Unified Modelling Language
XSL	: eXtensible Stylesheet Language
DTD	: Document Type Defination
CSS	: Cascading Style Sheets
HTTP	: Hypertext Transfer Protocol
CDATA	: Character Data
URI	: Uniform Resource Identifier
SAX	: Simple API for XML
DOM	: Document Object Model
VRML	: Virtual Reality Modeling Language
SOAP	: Simple Object Accessing Protokol
XMLMSG	: XML Messaging Protocol
RPC	: Remote Procedure Call
W3C	: World Wide Consortium
EDI	: Electronic Data Interchange
B2B	: Business to Business
SAVAGE	: Scenario Authoring And Visualization for Advanced Graphical Environments
OASIS	: Organization for Advancement of Structured Information Standards
IETF	: Internet Engineering Task Force
ASP	: Active Server Pages
IIS	: Internet Information Services
PDA	: Personel Digital Assistant

TABLO LİSTESİ

Tablo 3.1. DTD' de işaretleme ifadeleri	11
Tablo 3.2. DTD'de elemanları tanımlamada kullanılan semboller.....	13
Tablo 3.3. DTD'de özellik default tip tanımlama ifadeleri	14
Tablo 3.4. DTD'de özellik tipleri	14



ŞEKİL LİSTESİ

Şekil 2.1. XML belgeleri farklı ortamlarda erişilebilir	5
Şekil 2.2. Elektronik Veri Değişimi.....	6
Şekil 2.3. İç içe yuvalanma	8
Şekil 4.1. SAX API' nin XML belgesini parse etmesi	20
Şekil 4.2. DOM API' nin XML belgesini parse etmesi.....	21
Şekil 4.3. XML'in Stylesheet ile kullanılması	26
Şekil 4.4. XML'in XSL ile sunumu.....	27
Şekil 4.5. XML'in CSS ile sunumu	29
Şekil 5.1 SAVAGE projesi çıkarma hareketi Web3D görüntüleri	33
Şekil 5.2 SAVAGE Helikopter hareketi Web3D görüntüleri.....	34
Şekil 5.3. Harekat Sahasındaki Haberleşme Cihazlarının Yerleşimi ve Kapsama Alanları Görüntüsü.....	34
Şekil 5.4 ebXML'in temel yapısı.....	35
Şekil 5.5. JAXM temel yapı modeli.....	36
Şekil 5.6. SOAP mesaj formatı	36
Şekil 5.7 XML/EDI temel elemanları	37
Şekil 6.1 Çalışmanın Gerçekleştirildiği Bilgisayarın Donanım Özellikleri.....	38
Şekil 6.2 Giriş Ekranı Modülü	42
Şekil 6.5. Mesaj Okuma Ekranı	44
Şekil 6.6. Giden Mesaj Kutusu Modülü Ekranı.....	45
Şekil 6.7. Mesaj Gönderme Modülü Ekranı	46
Şekil 6.8. Üyelik Bilgileri Modülü Ekranı.....	47
Şekil 6.9. Online Hat Modülü Ekranı	47
Şekil 6.10. Üyelik Başvuru Modülü Ekranı.....	48

KARA KUVVETLERİ MESAJ SİSTEMİNDE XML TEKNOLOJİLERİNİN KULLANILMASI ÖZET

XML, HTML ile pek çok açıdan benzerlik gösteren bir işaretleme dilidir. Verinin tanımlanması ve tarif edilmesi için kullanılır. HTML'deki yapının aksine XML'de kullanılacak olan etiketler önceden tanımlı değildir. Yani bir XML dokümanının yapısı tamamıyla kullanıcı tarafından oluşturulur. Verinin tarif edilmesi için DTD adı verilen yapılar kullanılmaktadır. XML veriyi taşımak amacıyla tasarlanmış bir dildir. XML, yapısının esnekliği sayesinde birbirine uyumlu olmayan sistemler arasında veri alış verişini mümkün olabilmektedir

Bu tezde;

KKK' de birlikler arasında kullanılan mesaj sistemini XML tabanlı hale getirmek amaç edinilmiştir. XML'in özellikleri, yazım kuralları, kullanım alanları hakkında bilgiler verilmiştir. XML ile ilgili teknolojiler; DTD, XSL, XPath, DOM, XHTML vb. incelenmiştir. XML'in elektronik veri değişiminde standart olması için yapılan çalışmalar ile ABD Silahlı Kuvvetlerinin yürüttüğü XML tabanlı projeler araştırılmıştır.

KKK' de kullanılan mesaj formatı XML haline getirilerek mesaj alışverişinin XML olarak yapıldığı bir web uygulaması geliştirilmiştir. Uygulama; kullanıcı giriş kontrolünün yapıldığı üyelik modülüne, mesaj alışverişini sağlayan mesaj merkezi modülüne ve kullanıcılar arasında online iletişimi sağlayan online hat modülüne sahiptir. Modüllerin tamamı elektronik veri değişimini XMLHTTP protokolünü kullanarak XML formatında yapmaktadır.

Uygulamada, kullanıcı mesaj gönderme, mesaj okuma, mesaj silme işlemlerini gerçekleştirebilmektedir. Kullanıcıya, gelen yeni mesajlar vurgulanmakta ve gönderdiği mesajların okundu bilgisi geri bildirimi sağlanmaktadır. Gönderilen mesajlar kriptolanmakta (şifreleme) ve MD5 dijital imzalama tekniği kullanılarak imzalanmaktadır. Uygulamanın geliştirilmesinde Javascript kodlama ile CSS görüntüleme tekniklerinden faydalanılmıştır.

Bu çalışma sonucunda KKK'de kullanılan mesaj formatı XML'e dönüştürülmüştür. Böylelikle verileri her türlü sistem ve platforma taşımak mümkün hale getirilmiştir. Veriler XML dokümanlarında tutulduğundan herhangi bir veritabanı uygulamasına gereksinim kalmamıştır.

USING XML TECNOLOGIES ON TURKISH ARMED FORCES' MESSAGE SYSTEM SUMMARY

XML is a markup language which has many similar features with HTML. It is used for defining and describing data. Despite of HTML's structure, the tags used in XML is not described before. In other words, the structure of XML document is made by the user. The structures in DTD is used for defining data. XML is a standardized text format designed for transmitting structured data. Because of its flexibility of structure, XML makes the data exchange possible between incompatible systems.

To convert the message system, used between units on Turkish Armed Forces, to the XML based is aimed at this thesis. It was mentioned about the features of XML, its syntax and the areas that XML are used. XML related Technologies like DTD, XSL, XPath, DOM, XHTML was explained. The studies on XML for electronic data exchange to become a standard and XML based projects that are going on ABD Army forces was researched.

A web application which makes electronic data exchange as XML based was developed by changing the message format used in Turkish Armed Forces to XML.

Application has three modules that are logging which makes user verification, message center that exchanges messages between the users and online communication which the users communicate online. All of the modules exchange data by using XMLHTTP protocol and XML format.

In this application, user is able to sending, reading and deleting messages. The incoming messages were emphasized and the feedback knowledge of outgoing messages were given. The outgoing messages were ciphered and signed with MD5 which is a digital signature technique. JavaScript for coding and CSS for interface were used on development of the application

As a result, Turkish Armed Forces' message format was changed to XML format. So the exchange of data between different systems and platforms become possible. Because of that the data contained in XML documents there is no need to use a database system.

1.GİRİŞ

XML tam anlamıyla bir standart haline gelmemiş olmasına rağmen geleceğin web tabanlı uygulamalarında alt yapıyı oluşturacak en büyük aday olarak görünmektedir. XML, HTML ile pek çok açıdan benzerlik gösteren bir işaretleme dilidir. Verinin tanımlanması ve tarif edilmesi için kullanılır. XML'in açılımı Extensible Markup Language (Genişleyebilir İşaretleme Dili)'dir. XML'in özellikleri veri yapılarını, içeriklerini ve kavramlarını platform, şirket ve dilden bağımsız bir yapıda temsiline imkan vermektedir.

XML kısa zamanda kendisine birçok kullanım alanı bulmuş ve yazılım geliştiren firmalar tarafından benimsenmiştir. Yakın bir gelecekte tüm web uygulamalarının XML tabanlı olacağı değerlendirilmektedir. XML özellikle veri alışverişinde standart olarak kabul edilmesi için çalışmalar devam etmektedir.

XML'in sahip olduğu bu potansiyel birçok yabancı Silahlı Kuvvetlerinin ilgisini çekmiş ve XML'e dayalı projeler geliştirmeye başlamıştır. XML tabanlı veri alışverişi bize hız, esneklik, yalınlık, farklı platformlarda uygulanabilirlik sağlamaktadır. ABD Silahlı Kuvvetleri, hareket emirlerini XML tabanlı oluşturarak, hareket sahasında elde edilen bilgilerle hareket alanını 3 boyutlu görünümünü elde etmek amacıyla çalışmaktadır.

Milli güvenliğimizin teminatı olan Silahlı Kuvvetlerin mesaj sistemine aynı amaç ve ilkeler doğrultusunda;

- Birlikler arasındaki mesajlaşma trafiğinin; daha hızlı, esnek ve her ortamdan ulaşılabilir özellikler katmak,

- Özellikle karar verme durumunda olan kişilere bilgiyi uygun yer ve zamanda ulaştırabilmek maksadıyla,

bu çalışmaya başlanılmıştır. Uygulamanın geliştirilmesi aşamasında konularında uzman kişilerin görüş ve önerileri dikkate alınmıştır.

2. XML NEDİR

2.1 Giriş

XML, HTML ile pek çok açıdan benzerlik gösteren bir işaretleme dilidir. Verinin tanımlanması ve tarif edilmesi için kullanılır. XML'in açılımı Extensible Markup Language (Genişleyebilir İşaretleme Dili)'dir.

2.2 İşaretleme Dilleri Ve Tarihçesi

Metin ve belgelerin kolay bir şekilde taşınabilmesi, paylaşılabilmesi ve işlenebilmesi için ilk işaretleme dili GML (Generalized Markup Language) 1960 sonlarında IBM'de yapılan araştırma çalışmaları sonunda ortaya çıkmıştır. GML daha sonra ANSI (American National Standard Institute) 1978'te oluşturulan bir grup tarafından geliştirilerek SGML (Standardized Generalized Markup Language) adı altında 1986 yılında ISO (the International Organization for Standardization) kurumunca uluslararası bir standart olarak kabul edilmiştir [1,2]. SGML bir metin veya belge kümesinde kullanılan dilin gramer ve sözlük yapısını belirtmek için kullanılan bir dildir. SGML Amerikan hükümeti kuruluşlarında, havacılık ve otomobil gibi büyük endüstri kuruluşları ve basın endüstrisinde bir belgeleme standardı olarak kullanılmaktadır. SGML çok güçlü bir dil olmasına rağmen son derece karmaşık yapısı ve yüksek uygulama geliştirme maliyeti bu dilin yaygın bir şekilde kullanımını engellemiştir.

Tim Berners-Lee ve Anders Berlung 1989 yılında internet ortamında belge paylaşımını kolaylaştırmak için Web uygulamalarının temel öğelerinden biri olan HTML (Hypertext Markup Language) dilini geliştirdiler. HTML bir SGML uygulaması olarak geliştirildi. Diyebiliriz ki HTML dilinin yapısı SGML dilinde tanımlandı. HTML dili çok basit yapısı ile son derece başarılı oldu[3].

HTML dili bir belgenin içerdiği başlık, font, resim ve tablo gibi bilgileri bilgisayar ortamında standart bir şekilde görüntülemek ve biçimlendirmek için geliştirilmiş bir dildir. Belgenin istenen formatta sunulması etiket olarak ifade edilen işaretler ile sağlanmaktadır.

Kısa zamanda web uygulaması geliřtiren insanlar HTML'e yeni özellikler eklenmesini istediler. Bu da HTML'in řiddetli büyüme sorunları yaşamasına neden oldu. Sonrasında yanıp sönen metinler, çizelgeler, çerçeveler ve dinamik HTML ortaya çıktı. Her zaman HTML'e yeni bir şeyler ekleme ihtiyacı hissediliyordu fakat bu ekleme çabaları her defasında uyumsuzluk yaratıyor ve yeni standart ihtiyaçları ile sona eriyordu. Bu sorunların nedeni çok açıktı, HTML genişleyebilir değildi.

Bu ve benzeri sorunlar XML dilinin geliştirilmesine yol açtı. 1996 yılında World Wide Consortium (W3C),[4] SGML'in güç ve esnekliğini içerecek basit bir işaretleme dili oluşturmak amacıyla XML dilini tasarlamaya başladı. XML tasarlanırken ortada birçok düşünce vardı. W3C XML'in tasarım amaçlarını şöyle açıklamıştır.

1. XML Internet üzerinde çalışabilmelidir.
2. XML tüm uygulamalar tarafından desteklenmelidir.
3. XML, SGML ile uyumlu olmalıdır.
4. XML üreten programlar oldukça kolay olmalıdır.
5. XML' de seçmeli özellikler minimum olmalı veya hiç olmamalıdır.
6. XML belgeleri okunabilir ve açık olmalıdır.
7. XML tasarımı tek bir firma bu işi ele almadan acilen hazırlanmalıdır.
8. XML tasarımı biçimlendirilmiş ve kısa olmalıdır.
9. XML belgelerinin yaratılması çok kolay olmalıdır.
10. XML belgelerinde anlam belirsizlikleri olmamalıdır.

Şubat 1998'de XML 1.0 bir standart olarak W3C tarafından yayınlandı.

2.3 Neden XML

1. Açıklık: XML W3C standardıdır ve yazılım liderleri tarafından da onaylanmıştır.
2. Genişleyebilirlik: Sabit bir etiketler kümesi bulunmaz. Yeni etiketler istenildiğinde yaratılabilir.

3. Yalnlık: XML'de kodlanan bilgilerin okunması ve anlaşılması kolaydır çünkü doğal dile yakın anlamlı etiketler insanlar tarafından daha kolay anlaşılır.

4. Kendi kendini tanımlama: Alışlagelmiş veri tabanlarında ; veri kayıtları , veri tabanı sorumlusu tarafından kurulan şemalara ihtiyaç duyar. XML belgeleri bu tür tanımlamalara ihtiyaç duymadan saklanabilir çünkü XML etiket ve özelliklerden oluşan meta veriler içerir.

5. Makine tarafından anlaşılabilen anlam bilgisi içerir: Etiketler, özellikler ve eleman yapıları; içerik anlamını yorumlamada, etkin arama motorlarında yeni olasılıklar oluşturmada, akıllı veri işlemede vb. anlam bilgisi sağlar. Anlam bilgisinin değerlendirilmesinin zorluğu tartışılmayacak HTML veya düz metine göre bu büyük bir avantajdır.

6. İeriği gösterimden ayırır: XML etiketleri, gösterimi değil anlamı tanımlar. Bir XML belgesinin görünüşü ve anlattıkları XSL "stylesheet"leri veya CSS'ler tarafından oluşturulur. Bu, bir belgenin görünüşünün, belgenin bağlamına dokunmadan değiştirilebilmesini sağlar. Aynı içerikten ortaya çıkan birçok görünüm veya sunum kolayca oluşturulabilir.

7. Çok dilli belgeleri ve "Unicode"u destekler: Uygulamaların uluslararası hale getirilmesini sağlamaktadır.

8. Verilerin kümelenmesi ve karşılaştırılmasını kolaylaştırır: XML belgelerindeki ağaç yapısı, belgelerin eleman etkin bir biçimde kümelenmesine ve karşılaştırılmasına izin verir.

9. Çoklu veri türleriyle kullanılabilir: XML belgeleri, çoklu ortam verilerinden (resim, ses, video) aktif bileşenlere (Java Appletleri, ActiveX) kadar birçok olası veri türünü içerebilir.

10. Varolan veri türleriyle kullanılabilir: Varolan kütük dizgeleri veya ilişkisel veri tabanlarının XML'le ilişkilendirilmesi çok kolaydır.

11. Dağıtılmış veriler için tek bir sunucudan görünüş sağlar: XML'in erişim sağladığı birçok verinin öğeleri değişik veri tabanları içerisinde bulunabilir XML ile bu verilere tek bir sunucu üzerinden bakılıyormuş gibi erişilebilir.

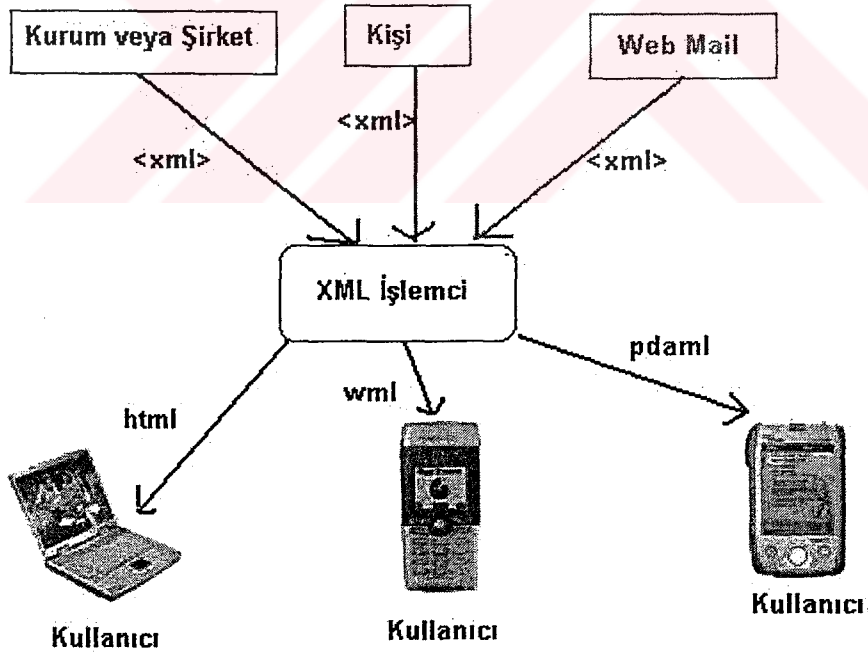
12. Yazılım geliştirici firmalar tarafından hızlıca benimsenmiştir: IBM, Sun, Microsoft, Netscape gibi birçok kurum XML'i desteklemektedir.[5-8]

2.4 XML Tabanlı Teknolojiler Ve Uygulamaları

XML'in başlıca kullanım alanları şunlardır;

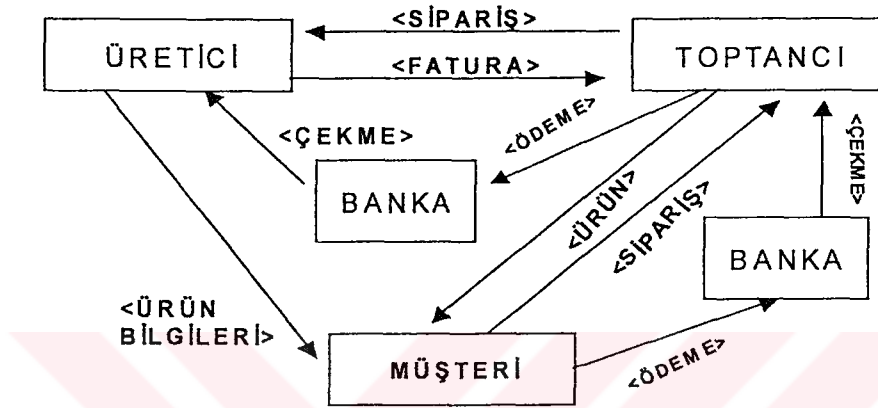
1. Internet Arama makineleri: Internet çok büyük bir hacimdeki yapısal olmayan veri içermektedir. Şu anda Internet belgeleri HTML ortamında saklandığından veri içeriği hakkında her hangi bilgi yoktur. XML yaygın bir şekilde Internet'te belge saklama ortamı olarak kullanıldığında, arama makineleri veriler hakkında verilen bilgileri kullanarak daha akıllı seçimler sunabilecektir.

2. Aygıt ve Uygulamadan bağımsız Veri Erişimi: Şu andaki Web ortamındaki HTML belgelerine masaüstü bilgisayarlardaki tarayıcı programları kullanarak erişilmektedir. Bunun yanı sıra cep telefonları ve kişisel sayısal yardımcı PDA (Personel Digital Assistant) gibi araçlar ile bugün web erişimi hızla artmaktadır. Bu araçlardaki görüntü, bant genişliği, bellek ve işlemci sınırlamaları nedeniyle HTML ihtiyacı karşılayamamaktadır. XML belge içeriğini belge sunumundan ayırmaktadır. XML ortamındaki belgeler aygıtın yapısına bağlı olarak istenilen şekilde formatlanarak sunulabilir Sekil 2.1.



Şekil 2.1. XML belgeleri farklı ortamlarda erişilebilir

3. Elektronik Veri Değişimi ve Elektronik Ticaret : XML'in en fazla ilgi çeken tarafı e-ticaret, tedarik zinciri yönetimi ve iş akışı yönetimi gibi elektronik iş web uygulamalarında bir veri değişim formatı olarak kullanılmasıdır. Bugünkü bilgisayar iletişimindeki çok büyük gelişimlere ve birikime rağmen, bilgisayar sistemlerindeki ve veritabanlarındaki farklı formatlardaki verilerin şirket içi ve şirketler arası taşınması ve işlenmesi en büyük problemlerden birisidir. Bu verilerin XML formatına dönüştürülerek taşınması bu problemin çözümünü kolaylaştırmaktadır.[1]



Şekil 2.2. Elektronik Veri Değişimi

4. Basın ve Yayıncılık: XML basın ve yayıncılık için geliştirilmiş olan SGML dilinin basitleştirilerek türetilmiş bir dil olduğundan bu konuda aynı amaçla kullanılabilir. XML formatında saklanan belgeler bir çok farklı ortama XML araçlarını kullanarak dönüştürülebilir ve kullanılabilir.

5.Şirket Uygulamaları Bütünleştirilmesi: Şirket Uygulamaları Bütünleştirilmesi kavramı farklı uygulamaların tek bir uygulama gibi davranması olarak ifade edilebilir. Bu işlem farklı sistemlerdeki veri kopyalarının ve dağınık bir yapıdaki verilerin doğru zamanda ve doğru yerde bulunmasını gerektirir.

6.Yazılım Geliştirme: XML büyük çaplı uygulama geliştirmede uygulama mimarilerinin paylaşımını kolaylaştıracaktır. XMI, UML (Unified Modelling Language) kullanarak tasarlanmış olan uygulamaları tanımlamada kullanılan bir dildir. XMI, geliştirme ekibinin çok farklı uygulama geliştirme araçlarını kullanarak, tek bir UML modelinin paylaşımına imkan vermektedir.

Bu yazılım geliştirme sürecinde geliştirme ekibi arasındaki iletişimi artırarak geliştirme sürecini kısaltmakta ve ekibi daha üretken kılmaktadır. Yazılım geliştiricileri XML meta bir dil olduğundan uygulamalarının gerektirdiği yeni dilleri geliştirebilir. Bu dillerde tanımlanmış olan verileri ve yapıları hazır olarak bulunan XML kütüphaneleri ve araçları ile kolaylıkla işleyebilir.

2.5 XML Söz Dizimi

XML belgeleri işaretleme ifadeleri ve veri içeriğinden oluşan bir veri dosyası veya paketidir. İşaretleme ifadeleri elemanlar (elements), varlık referansları (entity references), açıklamalar (comments), işleme komutları (processing instructions), CDATA bölümleri ve veri tipi tanımlamalarından (document type declarations) oluşur. [3,9-12]

2.5.1 Elemanlar

Elemanlar XML dilinin en temel yapı taşlarıdır. Bir eleman veri içeriğini kapsayan bir kutudur. Karakter verileri, diğer elemanları veya diğer işaretleme yapılarını içerebilirler. Elemanlar etiketlerle ayrılırlar. Her bir etiket bir eleman tip ismi içerir ve bir çift (“<” ve “>”) açılı parantezle içe alınır. Her bir isim bir harfle veya alt çizgi (_) başlayabilir.

Bir elemanın başlama ayırıcı başlangıç etiketi olarak adlandırılır ve bir çift (“<” ve “>”) açılı paranteziyle kapatılır.

```
<kime>  
<Kimden>
```

XML dili harflerin büyük veya küçük duyarlı olduğu unutulmamalıdır, <kime> ve <Kime> etiketleri aynı değildirler.

Bir elemanın bitiş ayırıcı bitiş etiketi olarak adlandırılır. Bitiş etiketi eleman tip ismi öncesinde bir bölü (“/”) karakteri içerir ve açılı parantezleri ile kapatılır.

```
</kime>  
</Kimden>
```

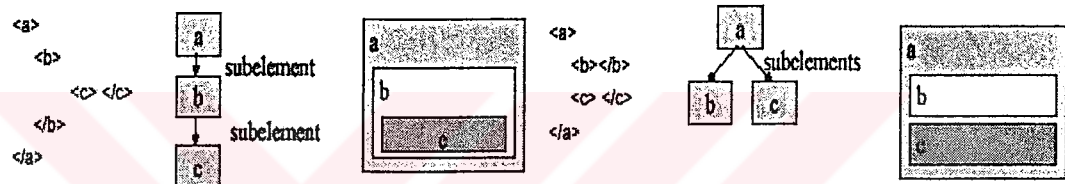
XML’de her bir başlangıç etiketi bir bitiş etiketi ile mutlaka kapatılmalıdır. Başlangıç bitiş etiketi ile kapatılmış bir tam XML elemanı aşağıdaki gibidir.

```
<kime>ÜÇORKOM</kime>
```

Aşağıdaki ilk satırda olduğu gibi boş elemanlar her hangi bir içerik içermezler. Boş etiketler istenirse ikinci satırdaki gibi etiketin sonuna tek bir bolü (/) işareti eklenerek kısaltılmış bir biçimde gösterilebilir.

```
</br></br>  
<br/>
```

XML belgeleri mutlaka hiyerarşik etiket yapısında olmalıdır. Etiket hiyerarşisi içindeki en üst seviyedeki etiket kök etiket veya belge kökü olarak adlandırılır. Kök etiketin oluşturduğu eleman Belge elemanı olarak adlandırılır. Bu eleman belgenin ağaç yapısındaki diğer bütün elemanları içine alır ve diğer bir elemanın içinde bulunamaz. Elemanlar iç içe yuvalandığında yuvalama kurallarına uymalıdır. Yoksa bu bir XML belgesinin hiyerarşik ağaç yapısını bozacaktır.



Şekil 2.3. İç içe yuvalanma

```
<B>  
  <I> ÜÇORKOM <B>  
    <I>  
HTML'de rasgele yuvalanma kullanılabilir  
<B>  
  <I> ÜÇORKOM <I>  
<B>  
XML'de sadece iç içe yuvalanma kullanılabilir
```

2.5.2 Özellikler

Elemanlara eklemek istediğimiz ek bilgiler özellikler olarak ifade edilir. Özellikler başlangıç etiketi içerisinde isim ve değer ikilileri olarak geçer.

```
<mesaj msgno="123456">
```

Buradaki msgno mesajın bir özelliğidir. Özellik değerleri mutlaka çift tırnak (") veya tek tırnak (') içine alınmalıdır.

2.5.3 Karakter Veri

Bir elemanın başlangıç ve bitiş etiketleri içinde geçen metin içeriği karakter veri olarak adlandırılır. XML içinde özel amaçlı olarak kullanılan (&,<,>,"') karakterleri olduğu gibi bir karakter veri içerisinde geçemezler. Bu karakterler işaretleme etiketleri içinde özel amaçlarla kullanılan ayraçlardır.

XML’ de HTML’den farklı olarak etiketler içine yazılmış olan veri içerisinde geçen boşluklar korunur ve uygulamalara iletilir. Etiket tanımları içerisinde veya özellik tanımı içerisinde geçen boşluklar ise HTML’de olduğu gibi yorumlayıcı (parser) tarafından yok edilir.

2.5.4 Karakter ve Varlık referansları

SGML ve HTML gibi ASCII karakter seti içinde karakterleri göstermek için XML iki yöntem kullanmaktadır. Birinci yöntem karakter referanslarıdır. Karakter referansları gösterilecek karakteri onluk bir sayının başına “&#”, onaltılık bir sayının başına ise “&#x” ekleyerek gösterme imkanı sağlar. Örneğin copyright karakteri ondalık olarak “©” veya onaltılık olarak “©” XML’de gösterilebilir.

Varlık referansları verilecek bir isim vasıtasıyla istenilen karakterleri göstermek için kullanılır. Varlık referansları bir amersen (&) karakteri, bir isim ve noktalı virgülden (;) oluşur. Örneğin XML’de özel olarak kullanılan (&,<,>,’,”) karakterleri (& < > ' ") varlık referansları ile gösterilir. Bu referanslar XML de daha önce tanımlanmış standart referanslardır. Bu referanslar haricinde özel olarak kullanmak istediğimiz varlık referansları DTD içinde tanımlanmalıdır.

2.5.5 Açıklamalar

Açıklamalar HTML’de olduğu gibi “<!--“ ile başlar ve “-->” ile biter. Açıklamalar bir XML belgesinin herhangi bir kısmında geçebilir ve belgenin bir parçası sayılmazlar.

2.5.6 İşlem Komutları

XML belgelerini kullanan uygulamalara her hangi bir komut geçilmesi istenildiğinde bu işlem komutları ile yapılır. İşlem komutları aşağıdaki yapıya sahiptir:

<?hedef ...komutlar ... ?>

İşlem komutları XML işlemcisi tarafından uygulamaya iletilir. İşlem komutları açıklamalar gibi bir XML belgesinin parçası sayılmazlar.

2.5.7 CDATA bölümleri

CDATA bölümleri XML’de kullanılan işaretleme ifadelerinin göz ardı edilmesi gereken metinleri yazmada kullanılır. XML yorumlayıcısı bu bölümde metin içerisinde yer alan işaretleme ayraçlarını ve etiketlerini normal karakter olarak algılar. Bir CDATA bölümü şu yapıya sahiptir:

<![CDATA [text]]>

3. XML VERİ TİPİ TANIMLAMALARI (DTD)

DTD'ler bir belgenin yapısını belirlerler. XML'in ilk harfinde geçen genişleyebilir kelimesinin (Extensible - X) anlamını ifade eden yeni yapılar tanımlamamıza imkan verir. Herhangi bir alandaki uygulamalarda kullanılacak yeni işaretleme dillerini (markup languages) DTD'ler ile geliştirebiliriz. Bu amaçla geliştirilmekte şu anda birçok XML işaretleme dili vardır. Bu diller XML uygulaması (application) veya XML sözlüğü (vocabulary) olarak anılmaktadır. XML sözlükleri bir alandaki bilgi değişimi ortamını sağlamak için tanımlanmış işaretleme dilleridir.

1. Synchronized Multimedia Integration Language (SMIL)
2. Chemical Markup Language (CML)
3. Bioinformatic Sequence Markup Language (BSML)
4. Common Business Library (CBL)

Bu yeni diller için tanımlanan DTD'ler bu dilin hangi etiketleri içerebileceği, etiketlerin sahip olabileceği özellikleri, hangi elemanların diğer hangi elemanları içerebileceği gibi dil yapısı bilgilerini (gramer) içerir.[3,9-13]

3.1 Belge Tipi Deklarasyonu

Belge tipi deklarasyonları bir XML belgesinin giriş kısmında tanımlanan belgenin gramer tanım kurallarını belirleyen hangi harici ve dahili DTD (Document Type Definiton)'yi kullandığını belirtir. Bu kısım varlık referansı tanımlamaları gibi ek bilgilerde içerebilir. Harici belge tipi deklarasyonları şu formatta olabilir.

```
<!DOCTYPE kök_eleman_ismi SYSTEM "sistem_tanımlayıcısı">  
<!DOCTYPE kök_eleman_ismi PUBLIC  
"public_tanımlayıcısı""sistem_tanımlayıcısı">
```

Bu tanımdaki kök_eleman_ismi belge DTD' sinde en üst seviyede tanımlanan kök elemanı belirtir. Sistem_tanımlayıcısı ise harici olarak kullanılan DTD' ye URI (Uniform Resource Identifier) bir referansı içerir. İkinci satırdaki formatta XML yorumlayıcıyı ilk önce public_tanımlayıcısı'nı kullanarak bir URI referansı oluşturmaya çalışır. Eğer bu mümkün olmazsa sistem_tanımlayıcısı'ndaki referansı kullanır.

```
<?xml version="1.0"?>
<!DOCTYPE kkkmsgsys SYSTEM "http://abc.com/kkkmsgsys.dtd"
[<!ENTITY amp "&#38;#38;"> ]>
```

İlk satır adres belgesindeki kök elemanın kkkmsgsys olduğunu belirtmektedir. Belgenin DTD'si http://abc.com/kkkmsgsys.dtd dosyasında saklanır. Bunun haricinde köşeli parantez içinde tanımlanan kısım ise varlık referanslarını tanımlamaktadır. Bu kısımda istenirse dahili belge tipi tanımlamaları da bulunabilir.

3.2 İyi Oluşmuş ve Doğru Belgeler

XML söz dizimi kurallarına uyan XML belgeler iyi-oluşmuş (Well-formed) belgeler olarak adlandırılır. Eğer belge için tanımlanmış olan DTD'de geçen gramer kurallarına uyuyorsa bu durumdaki belgelerde doğru (Valid) belgeler olarak adlandırılır.

3.3 DTD Söz Dizimi(Syntax)

Bir XML belgesinin yapısı DTD' de aşağıda verilen dört çeşit işaretleme ifadesi ile tanımlanır.

Tablo 3.1. DTD' de işaretleme ifadeleri

DTD Yapısı	İşlevi
ELEMENT	XML eleman tipi bildirimi.
ATTLIST	Belirli bir ELEMENT tipine bağlanabilecek özelliklerin ve bu özelliklerin alabileceği değerlerin bildirimi.
ENTITY	Yeniden kullanılabilir içerik bildirimi
NOTATION	"Parse" edilmeyecek dış içeriğin (resim dosyaları vb) biçiminin bildirimi.

3.3.1 Elemanlar

Elemanlar bir XML belgesinin en temel yapılarıdır ve ELEMENT etiketi ile tanımlanırlar. Dört çeşit eleman tipi vardır:

1. Empty (Boş) : Eleman herhangi bir içerik içermez. Elemanın özellikleri olabilir.

2. Element Only (Yalnızca Eleman) : Eleman yalnızca alt seviye (child) elemanları içerebilir.

3. Mixed (Karışık) : Eleman alt seviye elemanlar ve karakter veri karışımını içerebilir.

4. Any : Eleman DTD'in izin verdiği içeriği içerebilir.

•Boş (Empty) Elemanlar

Boş elemanlar aşağıdaki formatta tanımlanır.

<!ELEMENT Eleman İsmi EMPTY>

Boş eleman XML belgesi içerisinde özelliği yoksa aşağıdaki ilk iki satırdaki gibi geçecektir.

•Yalnızca Eleman İçeren (Element Only) Elemanlar

Bu tipteki elemanlar aşağıdaki formattadır.

<!ELEMENT Eleman İsmi İçerik-Modeli>

Bu yapıda İçerik-Modeli yalnızca bu elemanın altında kalacak alt elemanları içerebilir. Aşağıda verilen mesajlar kök elemanı mesajların tanımı aşağıdaki gibidir.

<!ELEMENT mesajlar kime kimden öncelik? konu? mesaj>

Burada kime, kimden ve mesaj alt elemanlarını mutlaka gösterilen sırada XML belgesi içinde geçmesi gerektiğini göstermektedir. öncelik ve konu elemanları ise opsiyoneldir. Bu elemanların opsiyonel olduğunu (?) sembolü göstermektedir. XML' de çeşitli amaçlarla kullanılan sembollerin anlamları Tablo 3.2. de verilmiştir.

Tablo 3.2. DTD’de elemanları tanımlamada kullanılan semboller

Eleman Tanımı	Anlamı
A?	Opsiyonel; A alt elemanı bir defa veya hiç gözükmebilir.
A+	A alt eleman mutlaka bir defa veya daha fazla gözükür.
A*	A alt eleman sıfır veya daha fazla gözükabilir.
A,B	B elemanı A’dan sonra gözükmelidir.
A B	A veya B’den bir tanesi gözükabilir.
()	Parantez içinde geçecek elemanları gruplar. Örneğin (A,B)+ ifadesi A ve B’nin ardışık olarak bir veya daha fazla geçeceğini göstermektedir

•Karışık (Mixed) Elemanlar

Karışık elemanlar hem karakter veri hem de alt elemanları içerebilirler. Karakter veri #PCDATA terimi ile gösterilir ve parsed karakter veri anlamındadır. Aşağıda verilenler gibi olabilir.

```
<!ELEMENT konu (#PCDATA)>
<!ELEMENT ekbilgi (#PCDATA | tarih | konu)*>
```

İlk satırdaki ifade konu elemanının yalnızca karakter veri içereceğini göstermektedir. Bu tip yalnızca veri içeren eleman tanımlarında #PCDATA parantez içine alınmalıdır. İkinci satırdaki karmaşık ekbilgi elemanın karakter veri, tarih veya konu elemanlarından oluşabileceğini göstermektedir.

•ANY Elemanları

ANY tanımının formatı aşağıdaki şekildedir.

```
<!ELEMENT Eleman İsmi ANY>
```

ANY elemanları herhangi bir yapıya sahip değildir. ANY elemanı karakter veri, tanımlanmış elemanlar veya bunların bir karışımını içerebilir.

3.3.2 Özellikler (Attributes)

Özellikler elemanlar hakkında ek bilgi tanımlamak için kullanılır. Bir eleman hakkındaki özellik bilgileri aşağıdaki formatta tanımlanır.

```
<!ATTLIST Elemanİsmi Özellikİsmi ÖzellikTipi Default>
```

Bir özellik bir isim ve tip bilgisini içerir. Default bir değer olabilir veya özellik kullanımı hakkında bilgi verir. Dört çeşit default değer tipi Tablo 3.3.te verilmiştir.

Tablo 3.3. DTD’de özellik default tip tanımlama ifadeleri

	Anlamı
#REQUIRED	Özellik her zaman tanımlanmalıdır..
#IMPLIED	Özellik opsiyoneldir.
#FIXED Değer	Özellik sabit bir Değer’e sahiptir. Özellik eleman içinde tanımlı olmazsa, parser bu sabit değer’in olduğunu varsayar.
Değer	Özellik eleman içinde tanımlı olmazsa, parser bu değer’in olduğunu varsayar. Eğer tanımlı ise özellik farklı bir değer içerebilir

Aşağıdaki ifadede Uye elemanının Adres diye bir özelliğinin olduğu ve bu özelliğin karakter veri olduğu (CDATA) belirtilmektedir.

<!ATTLIST Uye Adres CDATA #REQUIRED>

Bu özellik bir XML belgesi içinde şu şekilde geçebilir:

<Uye Adres="Atatürk mh.Seven sk.no:1"> </Uye>

XML 10 değişik özellik tipini destekler. Bu özellik tipleri Tablo 3.4. te verilmiştir.

Tablo 3.4. DTD’de özellik tipleri

Özellik Tipi	Anlamı
CDATA	Karakter veri (Unparsed)
Enumerated	Özellğin sahip olabileceği değerler kümesi
NOTATION	DTD’de tanımlanmış herhangi bir notation.
ENTITY	Harici bir varlık.
ENTITIES	Bir den fazla boşlukla (white space) ayrılmış varlıklar.
ID	Bir belge içerisinde geçebilecek tek tanımlayıcı bir değer (unique identifier).
IDREF	DTD’te tanımlanmış olan bir ID’ye olan referans.
IDREFS	Boşlukla (white space) ayrılmış ID’ler serisine referans.
NMTOKEN	XML token karakterlerinden (harf, sayı, nokta, tire, iki nokta, alt çizgi) oluşan bir isim.
NMTOKENS	XML token karakterlerinden oluşan boşlukla ayrılmış bir çok isim.

1. Enumerated özellikler: Bir dizi değer kümesi içerir. Özellik, bunlardan bir değeri alabilir. Aşağıda sınıf elemanının kuvvet özelliğinin kara, hava, deniz olabileceğini göstermektedir. Eğer bir kuvvet seçilmezse varsayılan olarak kara değeri alınır.

<!ATTLIST sınıf kuvvet (kara | hava | deniz) "kara">

2. Notation özellikler: XML harici veri tiplerini tanımlamakta kullanılır. Aşağıda resim formatının gif veya jpeg olabileceği belirtilmektedir.

<!ATTLIST resim format NOTATION (gif | jpeg) #REQUIRED

3. ID tipindeki özellikler: Değerleri bir belge içerisinde yalnızca bir defa geçebilir. Bu bize bir elemanı unique olarak tanımlamamızı sağlar. Uye adlı bir elemanımız olsun. Her Uye uye-no adlı bir özellik içersin. Her üyenin numarası farklı olacağından bu özellik ID olarak tanımlanır.

<!ATTLIST Uye uye-no ID #REQUIRED>

4. IDREF özellikler: ID tipindeki elemanlara referans vermemizi sağlar. Birlik adlı bir elemanımız olsun. Birlik ile Uye arasındaki bire bir ilişkiyi IDREF ile belirtebiliriz.

<!ELEMENT Birlik EMPTY>

<!ATTLIST Birlik uye-no IDREF #REQUIRED>

Bu bize birliğin personel bilgilerini uye elemanından elde etmemizi sağlar.

5. IDREFS özellikler: İki eleman arasındaki bire çoklu ilişkileri göstermekte kullanılabilir. Takım adlı bir elemanımız olsun. Bir takım bir çok personel içerecektir. Bu ilişkiyi şu şekilde belirtebiliriz.

<!ELEMENT takım EMPTY>

<!ATTLIST takım per-no IDREFS #REQUIRED>

6. Token özellikler: Token boşluk içermeyen bir değerdir. XML parser'ı bir özelliği token olarak işlemesi durumumda aradaki boşlukları (white space) tek boşluğa indirger ve baş ve sondaki boşlukları siler. Bu işlem özellik değeri normalizasyonu (attribute value normalization) olarak adlandırılır. Aşağıda bir çok token içeren bir NMTOKENS özellik tipi tanımı ve XML ifadesi verilmiştir.

<!ATTLIST personel eğitim NMTOKENS #IMPLIED>

<personel eğitim="ilkokul ortaokul lise ">

3.3.3 Varlıklar (Entities)

Varlıklar herhangi bir veri parçasına bir isim vererek bu veri parçalarına referans vermemizi sağlar. Varlık tanımlamaları bir defa tanımladığımız bir değeri bir çok yerde ismi ile referans vererek tekrar kullanabiliriz. Üç çeşit varlık tanımlaması vardır:

1.Genel Varlıklar (General Entities)

2.Parametre Varlıklar (Parameter Entities)

3.Karakter Varlıklar (Character Entities)

1. Genel varlıklar: Bir metin parçasına bir isim vererek bu isimle metni kullanmamızı sağlar. Aşağıda bir birlik ismi için yapılmış tanımlama gösterilmiştir.

```
<!ENTITY birlikadı "3 ncü Kolordu Komutanlığı">
```

Tanımlamayı aşağıdaki gibi ismin başına (&) harfi ve sonuna (;) ekleyerek kullanabiliriz.

```
<duyuru>&birlikadı; sitesinden başvuru formları indirilebilir. </duyuru>
```

Bu ifadedeki &birlikadı referansı tanımlamadaki metinle yer değiştirecektir.

Genel varlık tanımlamalarını harici bir dosyaya ulaşmak içinde kullanabiliriz. Aşağıdaki tanımlamanı readme.txt dosyasına referans tanımlamaktadır.

```
<!ENTITY lisans SYSTEM "http://www.abc.com/lisans.txt">
```

SYSTEM ve URL dosyanın bulunduğu yeri belirtmektedir.

2. Parametre Varlıklar (Parameter Entities): Genel varlık tanımlamaları XML belgesi içinde bir metin için tanımladığımız isimle belgenin içinde yer almasını sağlar. Parametre varlık tanımlamaları aynı işlevi bize DTD'nin içinde yapmamızı sağlar. Parametre varlık tanımlamaları aşağıdaki formattadır:

```
<!ENTITY % Varlıkİsmi VarlıkTanımı>
```

Aşağıdaki örnekte boyut isimli bir parametre tanımı yapılmıştır.

```
<!ENTITY % resimboyutu "width height">
```

Parametre varlıkları varlık ismi başına (%) sonuna (;) konarak DTD içinde referans edilir. Aşağıda boyut'a yapılmış olan referansları görmekteyiz.

```
<!ELEMENT uyeresmi (%boyut;)>
```

Eleman tanımında %boyut; isminin geçtiği yerlerde (width height) metni yer alacaktır.

3. Karakter Varlıklar (Character Entities): Karakter varlık referansları verilecek bir isim vasıtasıyla istenilen karakterleri göstermek için kullanılır. Karakter varlık referans tanımlamaları aşağıdaki örnekte olduğu gibi yapılır:

```
<!ENTITY copyright “&#169” >
```

Bu tanımlamada ASCII kodlamada 169 copyright karakterini (©) belirtmektedir. Karakter varlık referansları bir amersend (&) karakteri, bir isim ve noktalı virgülden (;) oluşur. XML metninde ©right; yazdığımız yerler © sembolü ile yer değiştirecektir.

3.3.4 Notation (Gösterim)

Notation tanımlamaları XML harici verinin ve bu harici ile ilişkilendirilecek bir yardımcı uygulamanın tanımlanmasını sağlar. XML harici verilerin işlenmesi, örneğin gif formatında tanımlanmış bir resim verisini, ancak bu veri formatları ile ilişkilendirilecek bir uygulama ile yapabilir. Aşağıdaki örnekte gif formatındaki tanımlanmış veriyi ilgili uygulamayla ilişkilendirmektedir.

```
<!NOTATION gif SYSTEM “irfanviewer.exe”>
```

Bu notation gif dosyalarının irfanviewer.exe uygulaması ile işlenmesi gerektiğini belirtmektedir.

4. XML İLGİLİ TEKNOLOJİLER

4.1 Ayırıştırma (Xml Parsing)

Bir XML belgesi bir bakımdan da yalın bir metin belgesi olarak görülebilir. Tam olarak bir XML kütüğü arka arkaya sıralanmış gerçek bir anlamı olmayan karakterler dizisidir. Bir metin düzenleyici, XML kütüğü yaratabilir ama onu bir dizi karakter olarak görür. Sıradan bir metin düzenleyici, XML belgesi içindeki sıralı karakterlerin içinde barındırdığı mantıksal yapı hakkında hiçbir şey bilmez.

XML iki bilgisayar arasındaki iletişimde kullanıldığı için, XML belgesinin mantıksal içeriğine yada “anlamına” erişmek önemlidir. Bu nedenle XML belgesi, bir dizi karakterden mantıksal bir yapıya çevrilmelidir.

Bir XML parser’ı ya da işleyicisi, mantıksal yapıya dönüştürme işlemini gerçekleştirebilir. XML’in amacı bilgisayardan bilgisayara iletişime yardım etmek olduğu için, bir XML parser’ı diğer bir yazılımın ya da bilgisayarın daha sonra işleyebileceği bir çıktı üretir.

Bir XML belgesini parse etmenin amacı diğer uygulamaların, XML belgesinin mantıksal yapısının içeriğine erişip inceleme yapabilecekleri arayüzler üretmektir.

Bir programdan, XML belgesi okumanın dört yolu vardır.

1. Kendinizin ürettiği bir arayüz yazılımı kullanmak,
2. SAX(the Simple API for XML) arayüzünü kullanmak,
3. DOM(the Document Object Model) arayüzünü kullanmak,
4. Bir API’ye sahip XML Parser’ı kullanmak

Temelde iki çeşit XML API’si (Application Programming Interface) bulunur: ağaç-tabanlı ve olay-güdümlü (tree-based/event-driven). Ağaç tabanlı API’ler XML belgesini bir ağaç yapısına dönüştürür ve bir uygulamanın bu ağacı dolaşmasını sağlar. DOM isimli API bu tür API’lere bir örnektir.

API kullanan uygulama, veri-merkezli değil de belge-merkezli çalışacaksa yani uygulama belge içindeki veriler yerine belgenin kendisiyle olan işlemlere daha çok ağırlık veriyorsa ağaç-tabanlı API'ler çok daha kullanışlıdır. Bununla beraber ağaç kullanmak sistem kaynaklarınızda ağır bir yük yaratabilir. Eğer belgenin tamamı yerine bir kısmı ile ilgileniyorsak bu kötü bir özellik olacaktır.

Diğer taraftan “Event-driven” API'ler parse edilme esnasındaki olayları (eleman başlangıç ve bitişi gibi) doğrudan uygulama programına aktarır. Böylece ağaç yapısına gerek kalmaz. Uygulama programı, değişik olayları(event) ele alabilmek için “handler” lar geliştirir. SAX bu tür API'ler arasında en çok bilinenidir.

4.2 SAX (Simple API for XML)

SAX, W3C tabanlı olmayan pek az XML tabanlı standartlardan biridir. XML-DEV mailing List'deki tartışmalarda ortaya çıkmıştır. SAX , DOM gibi ağaç tabanlı API'lere göre , XML belgesine daha basit ve daha alt düzeyli bir erişim sağlar. Hafızaya sığmayan çok büyük belgeleri parse etmek SAX ile mümkündür[3].

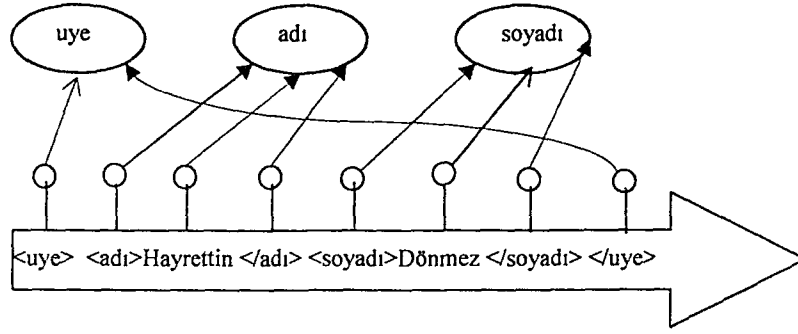
SAX 'ın çalışma prensibi şöyle açıklanabilir: Bir “XML Parser”ı XML belgesini taradığında SAX, her bir etiket ile bir olay ilişkilendirir. Bu olay'ın fark edilmesi uygulamaya bırakılır. Sonra da uygulamanın amacına göre her bir olay'ı ele alır. Kullanıcılar, SAX'ın ürettiği olaylar için “event handler”lar yazabilir. Gerekli işlemleri yapmak ve olayların , XML belgesi ile tutarlı olduğunu sınamak, olayı ele alan uygulamaya bırakılmıştır. Bu işlem için bütün belgeyi hafızada tutmaya gerek yoktur.Bunu bir örnekle açılırsak;

```
<?xml version="1.0"?>
  <uye>
    <adı>Hayrettin</adı>
    <soyadı>Dönmez</soyadı>
  </uye>
```

XML belgesi SAX tarafından aşağıdaki şekilde parse edilir.Belgeyi bir dizi olay haline getirir:

```
startDocument → startElement: uye → startElement: adı → characters: Hayrettin
→ endElement: adı → startElement: soyadı → characters: Dönmez →
endElement: soyadı → endElement: uye → endDocument
```

Bu olay dizisi Şekil 4.1. de görülebilir.



Şekil 4.1. SAX API' nin XML belgesini parse etmesi

SAX'a istemci konumunda olan bir uygulama bu olayları ele alır ve istenilen faaliyetleri yerine getirir. Bütün belgenin hafızaya alınmasına böylece gerek kalmaz ve belgenin içinde ilgilenmediğimiz herhangi bir bölümü yazacağımız uygulama ile ele almak zorunda kalmayız.

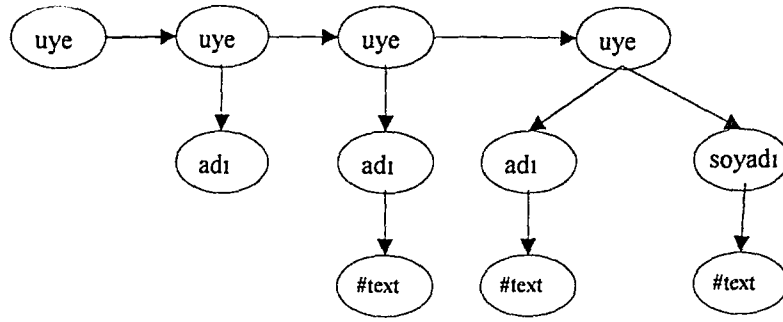
4.3 DOM (Document Object Model)

DOM, W3C tabanlı bir modeldir. Bu model ile XML belgesi düğümlerden oluşan bir ağaca dönüştürülür ve bu ağaç bilgisayar hafızasına alır. Bütün belge, hafızada bir kerede tutulmak zorundadır. Bu nedenle, çok büyük XML belgelerini DOM tabanlı yaklaşımla ele almak, hafıza yetersizlikleri nedeniyle kısıtlamalar getirebilir. Ayrıca DOM'u çok büyük belgelerde kullanmak, yavaş bir işlemdir.

DOM, XML ağaç yapısına ulaşımı kolaylaştırmak için bir dizi arayüzler içerir. Bu arayüzler de düğüm tipine göre değişen metot ve özellikler içerir. DOM'un temel arayüzü düğümdür. Herhangi bir düğüm ise eleman, özellik veya metin olabilir. Düğüm arayüzü bütün düğüm tipleri için ortak olan metot ve özellikler içerir. Diğer DOM arayüzleri ise düğümden türetilmişlerdir[3].

```
<?xml version="1.0"?>
  <uye>
    <adı>Hayrettin</adı>
    <soyadı>Dönmez</soyadı>
  </uye>
```

örnek XML belgemizi DOM, Şekil 4.2. de olduğu gibi parse eder ve nesne ağaç modeli oluşturur.



Şekil 4.2. DOM API' nin XML belgesini parse etmesi

Bugün internet tarayıcı programların çoğunluğu DOM API yi programlarının içine gömülmüş olarak desteklemektedir. Bu da JavaScript kodları ile DOM API' nin özelliklerine ulaşmayı mümkün kılmaktadır.

4.4 XML Parser'ları (XML Ayrıştırıcıları)

Bir XML Parser'ı (XML işleyicisi olarak da bilinir), XML belgesinin söz dizimini ve mantıksal içeriğini denetler. Bütün ayrıştırıcı XML belgesinin iyi-oluşturulmuş (well-formed) olup olmadığını kontrol eder Ek olarak, geçerleyici ayrıştırıcılar belgenin uygun DTD ile eşleşip eşleşmediğini doğrular.

Parser ne yapar? Bir XML Parser'ı XML metin kütüğünü sıradüzensel bir şekilde tarar. XML belgesinin elementleri gibi bölümleri belirler. Eğer Parser geçerleyici ise, XML belgesinde kullanılabilecek yapıyı belirlemek için, DTD üzerinde basit bir işlem yürütür. Daha sonra Parser, XML belgesinin yapısı ile DTD'ye uygunluk sağlayan yapıyı karşılaştırır. Bu işlem sonucunda da geçerlilik kontrol edilmiş olur.

Çok sayıda XML ayrıştırıcısı ücretsiz olarak kullanılabilir. Bunların çoğu hem DOM hem SAX modelini destekler. Bazıları şunlardır:

1. Project X (Sun Microsystems)
2. Ælfred (Microstar Software)
- 3.XML4J (IBM) Project X
4. MSXML (Microsoft)
5. Xerces (Apache)

4.5 XPath

XML belgesinin ağaç yapısı içerisindeki elemanların adreslenmesi için geliştirilmiş bir dildir. XPath, XML belgelerindeki elemanlar, özellikler ve diğer XML düğümlerinin adreslenmesi için basit ve standart bir yapı sağlar.

XPath'in, bir belge içindeki elemanlarının dizilişini gösteren şablonları sağlamak amacıyla da bir sözdizimi bulunur. Önemli noktalardan biri de şudur ki XPath XML sözdizimini kullanmaz. XPath'in tanımında XML sözdiziminin kullanılmamasının getirdiği yararlarından biri de XPath'in URI'ler ve XML özellik değerleri içinde kullanılabilmesidir.[14]

XPath XML'deki "namespaces" leri destekler. Çünkü XPath, XML belgesini düğümlerden oluşan bir ağaç olarak görür. Bir düğümün adı yerel bölüm ve "namespace" den oluşur. İkisi birlikte geniş ismi ("expanded name") oluşturur.

XPath sözdizimini oluşturan temel yapı, deyimleridir. XPath deyimleri, nesneleri işler ve sonuçta; "node-set" (birkaç düğümün birleşimi) , "boolean" (mantıksal) , "number" (sayı) veya "string" (dizi) türlerinden biri şeklinde sonuç üretir.

XPath'e göre, XML belgesinin ağacının düğümleri 7 türde olabilir. Bunlar :

- 1.Kök düğümler (root nodes) : Ağacın kök düğümü.
- 2.Eleman düğümleri (element nodes) : Belgedeki her eleman için oluşturulan düğümler.
- 3.Metin düğümleri (text nodes) : Karakter bilgileri metin düğümleri içerisine gruplandırılmışlardır.
- 4.Özellik düğümleri (attribute nodes): Her eleman düğümünün sahip olduğu özelliklerin her biri için bir çocuk bulunur.
- 5.Namespace nodes: Her eleman, bir küme namespace düğümü ile ilişkilendirilir.
- 6.İşlem komutları düğümleri (processing instruction nodes) : DTD (decleration) içinde yer alanlar hariç her işlem komutu bir düğümdedir.
- 7.Yorum düğümleri (comment nodes): DTD (decleration) haricinden her yorum satırı bir düğümdedir.

XPath'deki adresleme mekanizmasının kalbi, "location path" dir. "Location path"ler ulařılmak istenen yerin bulunulan noktaya (context node) göre yerini belirler. XPath'de 13 adet axis (eksen) bulunur:

1.child axis: Bulunulan düğümün çocuk düğümü. Bir düğüm için varsayılan (default) axis budur.

2.descendant axis: Bulunulan düğümün çocuğı veya çocuğunun çocuğı veya bu şekilde herhangi bir seviye altındaki alt düğüm.

3.parent axis: Bulunulan düğümün bir seviye üstteki düğümü. ".." ile ifade edilir.

4.ancestor axis: Bulunulan düğümün atası veya herhangi bir seviye üstündeki düğümü

5.following-sibling axis: Bulunulan düğümün devamındaki kardeş (aynı seviyede) düğümler

6.preceding-sibling axis: Bulunulan düğümden önce gelen kardeş düğümler

7.following axis: Belgedeki sıraya göre bulunulan düğümden sonra yer alan, alt düğümleri hariç bütün düğümler

8.preceding axis: Belgedeki sıraya göre bulunulan düğümden sonra yer alan, ataları hariç bütün düğümler

9.attribute axis: Bulunulan düğümün özellikleri. "@" öneki ile tarif edilir.

10.namespace axis: bulunulan düğümün namespace düğümleri

11.self axis: bulunulan düğümün kendisi. "." ile ifade edilir.

12.descendant-or-self axis: bulunulan düğümün kendisi ve ata (descendant) düğümleri. "/" ile ifade edilir.

13.ancestor-or-self axis: bulunulan düğümün kendisi ve alt (ancestor) düğümleri.

Ařağıda bazı XPath deyimlerinin kullanımı gösterilmiştir.

child::* :Bulunulan düğümün bütün çocuklarını seçer.

child::text() : Bulunulan düğümün çocukları arasından metin düğümlerini seçer.

child::node() : Bulunulan düğümün hangi türde olursa olsun bütün çocuklarını seçer.

child::uye[position()=1] : Bulunulan düğümün ilk uye çocuğunu seçer.

child::uye[position()=last()] : Bulunulan düğümün son uye çocuğunu seçer.

4.6 XPointer

XPointer'ın amacı, herhangi bir URI(Uniform Resource Identifier) referansı için, XML belgesinin içinde kesimler yaratmaktır.XPointer, XML belgesinin iç yapısını ve bu belgelerin içindeki XML kesimlerinin yerlerini adreslemeyi destekler. Diğer bir deyişle, XPointer sıradüzensel bir belgenin yapısının sıranmasına izin verir.[11]

XPointer, XPath üzerine kurulmuştur ve XPath'ın ifadelerini kullanır, ayrıca XPath'ın içindeki bazı ifadelerin genişlemesini sağlar. Bu genişlemeler XPath'in şunları yapmasına yardımcı olur:

Düğümün yanında XML belgesi içinde herhangi bir noktanın ya da herhangi iki nokta arasının da adreslenmesini sağlar

Dizi eşlemesi yaparın bilgilerin yerini belirler.

URI referanslarındaki adresleme deyimlerini kullanmaya olanak sağlar.

XPointer' da başlıca kullanılan deyimler şunlardır.

- #element : XML elemanlarını seçmekte kullanılır.
- #pi : XML işleyici komutlarını seçmekte kullanılır.
- #comment : XML açıklama satırlarını seçmekte kullanılır.
- #text: Elemanlar arasındaki ve CDATA bölümlerindeki metin alanlarını seçmekte kullanılır.
- #cdata: CDATA bölümleri içerisinde bulunan metin alanlarını seçmekte kullanılır.
- #all : Yukarıdaki düğüm türlerinden hepsi için kullanılır.

Bazı kullanım örnekleri aşağıdadır.

child(1,#element,adı,*) : adı özelliğine sahip ilk çocuğu seçer.

id(231454).child(4,#element): id özelliği '231454' olan elemanın 4 ncü çocuk elemanı seçer.

XPointer, XPath'e nazaran daha kuvvetli özelliklere sahiptir fakat birçok parser hala Xpointer'ı desteklememektedir.

4.7 XLink

XLink, hem temel bağlar; hem de daha karmaşık bağ yapıları için temel bir yapı oluşturur. Daha karışık bağ işlevlerinden hareketle, XLink, internet bağları (hyperlink) de sağlar. W3C'ye göre, XLink, XPointer ile koştur kullanılmalıdır.

XLink , XML belgelerine , kaynaklar arası bağları tanımlayan ve yaratan elementler yerleştirmeye izin verir. HTML'deki <a> etiketi gibi bağlar yanında XLink daha karmaşık bağlama işlevlerini de gerçekleştirir. XLink, kaynaklar arasındaki net ilişkidir. Bir kaynak (resimler, kütükler, programlar, belgeler ve sorgu sonuçları) ise bir bilginin ya da servisin herhangi bir adreslenebilir bileşenidir.[4]

Özetle XLink , XML dokümanlarına şunları yapmamızı sağlar:

- İki'den fazla kaynak arasında bağ kurmamızı,
- Meta-veriyi bağlarla birleştirmemizi,
- Bağlanmış kaynaklardan farklı yerlerdeki bağları ifade etmemizi sağlar.

XPointer gibi XLink te birçok parser tarafından hala desteklenmemektedir.

4.8 XML Şema (XML Schema)

XML Schema tıpkı DTD lerde olduğu gibi belge içindeki elementlerin ve içerik modellerinin tanımlanması işlemini ve dolayısıyla belgenin geçerlenmesini sağlar. Ancak XML Schemalar elemanların bu tanım işleminde zengin bir veri tipi çeşitliliği ve özellik desteği sağlamaktadır.

DTD içerisinde, eleman içeriği karakter ve diğer birkaç veri tipi ile sınırlıdır. XML Schema çok geniş bir veri tipi desteği sağlamaktadır (integer, floating point sayılar, date, time, gibi). XML Schema ayrıca açık içerik modeli(open content model) ve isim uzayı(namespace integration) gibi ek özellikler de sunmaktadır. XML Schema belgeleri oluşturulması sırasında XML sözdizimi kullanılmaktadır. XML Schema'nın bazı özellikleri aşağıda verilmiştir.[14]

- XML Schemalar XML sözdizimini temel olarak alır.Özel bir sözdizimi gerektirmez Diğer XML belgeleri gibi parse edilip üzerinde işlem yapılabilir.
- Çok geniş bir veri tipi desteği sağlar.

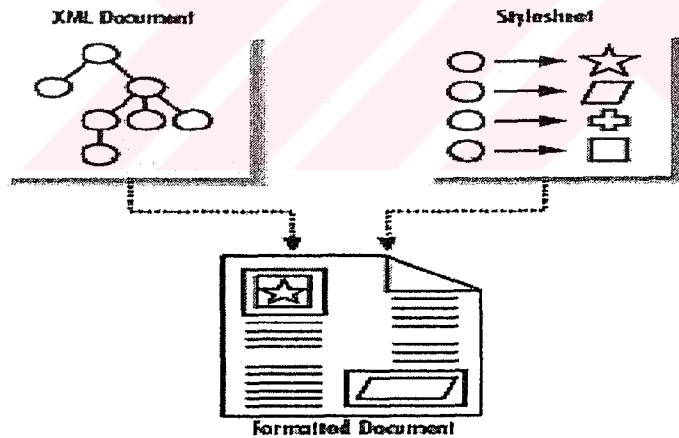
- Open-ended bir veri modeli sunar.Bu model sayesinde elemanlar arasındaki ilişkilerin kurulması ve vocabulary haznesinin genişletilmesi sağlanır.
- Namespace desteği sağlar.Bu özellik Schema içerisindeki bildirimler için ayrı node'ları birleştirmeye yarar.
- Özellik gruplarını destekler. Böylece özellikler mantıksal olarak gruplanabilir.

XML Schemalar DTD yapısının yeterli olmaması sonucu ortaya çıkmıştır.

4.9 XML Belgelerinin Sunumu

XML önceden tanımlanmış etiketleri kullanmadığı için (kendi etiketlerimizi tanımlayabildiğimiz için) , bu etiketlerin anlamı anlaşılmaz. <table>, bir HTML çizelgesi anlamına gelebildiği gibi başka birçok anlama da gelebilir. XML'in doğal yapısından dolayı tarayıcı XML belgesini nasıl görüntüleyeceğini bilemez.

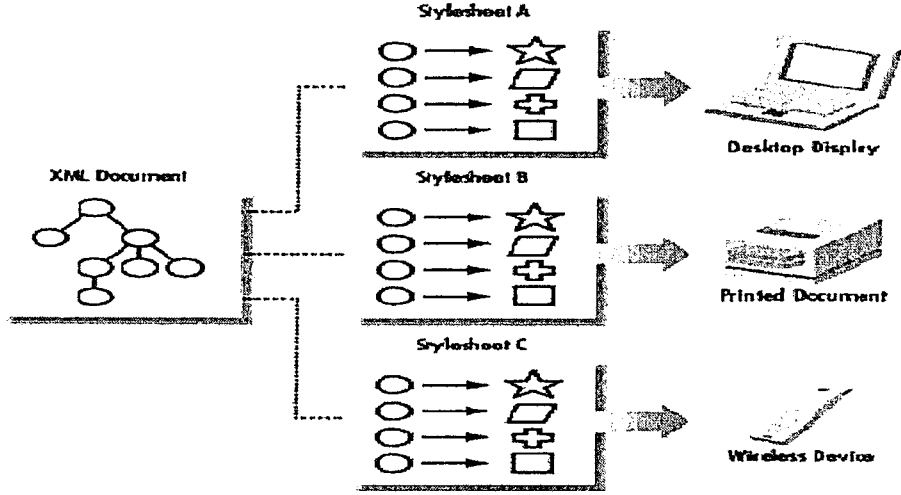
Bu XML belgelerini göstermek için, belgelerin nasıl gösterilmesi gerektiğini tanımlayan bir mekanizmaya ihtiyaç vardır. Bunlardan biri CSS (Cascading Style Sheets), diğeri ise XSL (eXtensible Stylesheet Language)'dir.[9]



Şekil 4.3. XML'in Styleshett ile kullanılması

4.9.1 XSL (eXtensible Stylesheet Language)

XSL, XML belgelerinin biçimlendirilmesi tasarlanmış bir biçimleme dilidir. CSS'den çok daha güçlü bir yapıya sahip olmasına rağmen XSL yeni bir teknoloji olduğundan çok fazla yaygın değildir ve konudaki standartlaştırma çalışmaları henüz tamamlanamamıştır. XSL de , XML'in yapısını kullanarak sunumlar hazırlamak için kullanılır. Fakat sunum, yalnız web üzerinden değil, aynı zamanda işitsel ya da yazılı da olabilir.



Şekil 4.4. XML'in XSL ile sunumu

XSLT, bir XML belgesinin dönüşümü için ilk önce belgeyi parse ederek bir ağaç yapısı oluşturur. Oluşan ağaç yapısının kök elemanından başlayarak ağacı tarar ve XSL biçimleme dosyasında tanımlanan kalıplara göre dönüşüm işlemini gerçekleştirir. XSLT ağaç yapısını tararken XPath ifadelerini kullanarak ağacın değişik kesimlerine erişir.

XSL 3 kısımdan oluşur:

1. XML belgelerini çevirmeye yarayan yöntem.
2. XML'in bölümlerini ve şablonlarını tanımlamaya yarayan yöntem
3. XML belgelerini biçimlendiren yöntem.

XSL, XML'i HTML'e yada başka yapıdaki XML belgelerine çevirmeye yarayan , XML verisini süzen ve sıralayan , XML belgesinin bölümlerini adresleyen , XML verilerini veri değerine göre biçimlendiren, XML verilerinin çıktısını değişik aygıtlara veren (ekran, kağıt vb.) bir dildir. Eğer belli yapıda bazı verilerimiz var ve aynı amaç içinde kullanmamız gereken başka yapılarda verilerimiz de varsa, ikisinin ortak bir yapıya dönüştürülmesi için, belgemizi başka bir XML belgesine çevirmek gereğini duyabiliriz.

Üstteki üç madde aslında XSL'i oluşturan üç ayrı dile denk düşer:

1. XSLT : XML'i çevirir.
2. XPath : XML'in bölüm ve şablonlarını tanımlar
3. XSL-FO(Formatting Objects) : XML görüntülenmesini tanımlar.

XPath spesifikasyonu oluřan bir XML ağacının içindeki düğümleđi adreslemek için kullanılır ve ağacın belirli kısımlarını adresleyerek ağaç içinde dolařmamızı saęlar

XSLT, XML belgelerini dięer türden belgelere veya dięer XML belgelerine çevirmeye yarar. XPath, XML belgelerinin kısımlarını adreslemekte kullanılır. XPath, XSLT tarafından kullanılır. Biçimlendirme işleđi, XSL çevirisinin sonuçlarının, kullanıcı (okuyucu yada dinleyici) tarafından anlaşılabilen uygun bir biçime dönüřtürölmesidir.

4.9.2 CSS (Cascading Style Sheets)

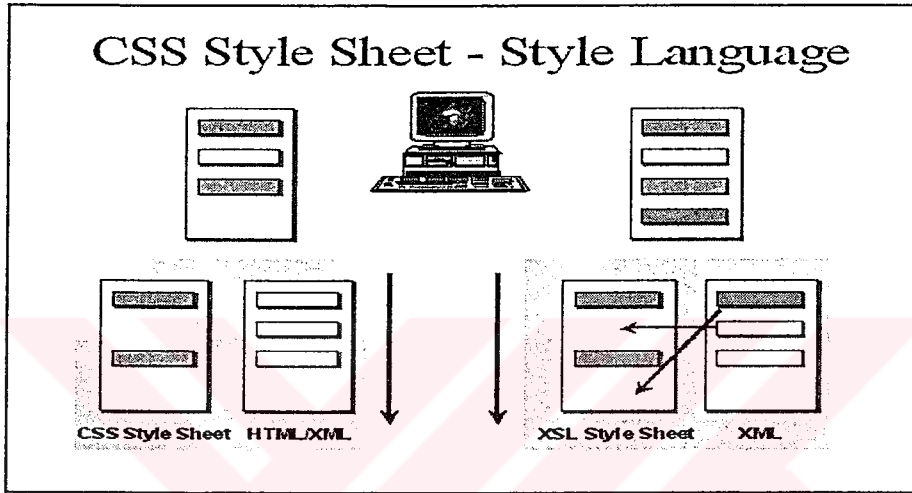
CSS HTML ve XML belgelerini görüntölleme amacıyla kullanılan bir biçimleme dilidir. CSS biçim sayfaları (style sheets) belge elemanlarına uygulanacak olan bir kurallar kümesi içeriđ. Her bir kural elemanların nasıl biçimlendirileceğini belirler. Bu tekniğin en önemli özellięi kullanımındaki esneklięidir. Bir web sayfası içerisinde estetik kuralları gereęi yüzlerce renk ve font kullanmayız. Genelde birbiriyle uyumlu birkaç renk ve birkaç font kullanırız ki, bunları her sayfada ayrı ayrı tekrar belirtmek yerine CSS yardımıyla bir sefer tanımlayıp bütün web sayfamızda kullanabiliriz. Bu şekilde güncelleme yaparken de onlarca sayfayı deęiřtirmekten kurtuluruz. CSS türleri üç tanedir:[15]

- Yerel stil řablonu: Sadece sayfa içerisinde tanımlandığı yerde geçerlidir.
- Genel stil řablonu: Tüm sayfayı kapsar.
- Harici stil řablonu: '.css' uzantılı bir dosya olarak kaydedilir, kullanılacağı sayfalarda çağırılmak suretiyle tüm web sitesini kapsayabilir.

Harici Stil řablonları CSS 'in bizi büyük bir yükten, yani her sayfa için ayrı ayrı kod yazmaktan kurtaran türü, harici stil řablonlarıdır. Bu teknikte önce bir stil dosyası oluşturulur, sonra bu stilin kullanılacağı sayfaların başında çağırılır.

CSS, bir Web dokümanın sunuşunu (fontlar, renkler, kenar boşlukları,...) , onun yapısına dokunmadan kontrol etmeye, deęiřtirmeye imkan saęlar. İşlem sırasında baz olarak doküman alınır.

XSLT'de ise sunuşun nasıl olacağını tamamen stylesheet içindeki komutlar belirler. Stylesheet, XML dokümanı içeriğini seçer ve sunum için biçimlendirmeyi belirler, hatta XML dokümanında hiç olmayan bilgiler de ekleyebilir. Yani işlem sırasında baz olarak stylesheet alınır. XSLT içinde, bir XML dokümanının bölümlerini tarif etmek için yine bir W3C standardı olan XPATH dili kullanılır. Ek olarak XSLT ile bir XML dokümanının belli bölümleri filtrelenebilir, bölümler sıralanabilir (sort), doküman tamamen dönüştürülebilir



Şekil 4.5. XML'in CSS ile sunumu

CSS basit biçimlendirme işlemleri için iyi bir yöntemdir. Karmaşık biçimlendirme işlemleri için XSL dili tercih edilir.

4.10 XQuery

XML Query SQL'in ilişkisel veri tabanlarının bağlamında kullanıldığı gibi kendi deyimleriyle XML belgelerine veya veri kaynaklarına erişim , sorgulama , kullanımı sağlar.[4]

XML Query'nin şekillenmesi halen devam etmektedir. Şu ana kadar üç düşünce şekil almıştır: veri modellemesi, sorgu cebiri ve sözdizimi. Buradaki asıl amaç XML Query'nin bildirilebilir ve sözdiziminin XML uyumlu olmasıdır. Ayrıca sıradüzenselliğe ve dizilere erişime, birleşikliğe izin verir.

Gittikçe daha çok bilgi XML kullanılarak saklanıp , transfer edildikçe XML veri kaynaklarını akıllıca sorgulamak da o denli önem kazanmaktadır. XML'in en büyük güçlerinden biri de çeşitli kaynaklardan elde edilen değişik bilgilerin sunumundaki esnekliğidir. Bu esnekliği iyi bir şekilde kullanabilmek için , XML sorgu dili , bu bilgileri almak ve yorumlamak için çeşitli özellikler sağlamalıdır.

XQuery aşağıdaki özelliklerle yakından ilgilidir :

- XQuery veri modeli , bir XML belgesi tarafından XQuery işleyicisine sunulmuş bilginin nasıl olması gerektiğini tanımlar.
- XQuery'nin tür dizgesi XML Schema'yı taban alır.
- XQuery oluşturmundaki diğer bir gerekli özellik , oluşturulan XML sorgu dilinin hem okunabilir bir söz dizime sahip olması hem de XML tabanlı bir söz dizime sahip olmasıdır.

XQuery güçlü bir dildir. Sorgu, bir şemadan tür tanımları alabilir. Bir sorguyu işletmek, durağan yazım aşaması ve dinamik hesaplama aşaması içerir. XQuery çoklu uyumluluk düzeyine sahiptir. Örneğin durağan yazım denetimini içermeyen bir uyumluluk düzeyi olabilir. Yada şemadan tür tanımları alımını desteklemeyen bir uyumluluk düzeyi de olabilir.

XQuery deyimlerinin kullanımına bir örnek aşağıda verilmiştir.

```
FOR $b IN document("msg.xml")//mesaj
WHERE $b/gizli = "özel" AND $b/öncelik = "normal"
RETURN $b SORTBY (msgid, kimden)
```

'msg.xml' belgesinde, gizliliği özel, önceliği normal olan mesajları mesajid'sine ve kimden olduğunu sıralayarak bul ifadesine eşittir.

4.11 XHTML

XML ilk piyasaya çıktığında, onun HTML'in yerini alacağını düşünen çok olmuştur. Aslında amaç her zaman için HTML'i XML kurallarına göre yeniden düzenlemektir. W3C bunu gerçekleştirmiş ve sonuçta XHTML oluşturmuştur. XHTML Ocak 2000'de bir W3C standardı haline gelmiştir.[4] Tarayıcı şirketleri bundan sonra tarayıcılarını XHTML kodu ile çalıştırılabilir hale getirmek için çalışmalarını hızlandırmışlardır.

XHTML gerçekte HTML'e çok benzemektedir. Böylece çok fazla yeni özellik ve element öğrenmeye gerek yoktur. Uygulama geliştiriciler için en önemli değişim, XHTML'in sözdizimi kurallarına sıkı sıkıya bağlı olma zorunluluğudur. Eğer söz dizimi hatalı bir HTML belgesi tarayıcıda gösterirseniz çoğu tarayıcı bunu görmemezlikten gelerek belgeyi yorumlayacaktır. Bu çeşit "görmezden gelme" özellikleri XHTML'de yoktur.

XHTML, HTML'in içerdiği birkaç sorunu ortadan kaldırır. Birincisi ve en önemlisi , bir XHTML belgesi sözdizimsel hatalı ise kullanılamaz. Bu XML sayesinde kazanılmış bir özelliktir. XML'in en önemli özelliklerinden biri de , bir belgenin sözdizimsel olarak doğru ("well-formed") veya DTD'sine uygun (geçerli "valid") olmadan uygulanamamasıdır. Bundan yola çıkarak eğer herhangi XHTML belgesi sadece bir sözdizimsel hataya bile sahipse reddedilir.

İkinci olarak, Web programcılarının CCS kullanarak, içeriğin web'de nasıl temsil edileceğine etki etmelerini sağlar. En kesin haliyle XHTML, herhangi bir elementinin, belgenin içeriğinin nasıl temsil edileceğine etki etmesine izin vermez.

XHTML'in başlıca özellikleri şunlardır.

- Standartlaştırma: HTML her zaman standartlaştırılmış bir dil olmuştur. Fakat bu standart, çoğu kez tarayıcılar ve yazılım araçları üreten yazılım şirketleri ve programlamacılar tarafından görmezden gelinmiştir. Netscape ve Microsoft'un hazırladığı patentli etiketler, diğer tarayıcılar için sorunlar çıkarmış ve W3C'yi HTML dünyasına bir düzen getirmeye mecbur etmiştir. XHTML ile herkes sözdizimsel kurallara uymak zorundadır. Böylece koddan doğan sorunlar kalkmış olacaktır.
- Genişleyebilirlik: XHTML bir XML uygulaması olduğundan dolayı, kalıtsal olarak genişleyebilir. Bunun anlamı diğer elementlerin sözdiziminde en küçük etkiyi yaratarak yeni elementlerin ve özelliklerin dile eklenebilmesidir. Bu XHTML'in sabit hızla değişen web yayını ihtiyaçlarına cevap verebilmesinin sağlayacaktır.
- Dağınık kodlamayı engeller: Kodu kodlama alışkanlığı, geçmişteki sitelerin kabusu haline gelmiştir. XHTML bizi iyi kod yazmaya zorlarken, işlerin kodla uğraşacak insan ve programlar için daha kolaylaşmasını sağlamış olur.

- Güçlendirilmiş bağlar: XML standartları, HTML standart internet bağlarından (hyperlink) çok daha iyi bir bağlama şekli sağlar. Örneğin, XML bağlamadaki bir fikir, tek bir internet bağının bir yerine birden çok kaynağa işaret etmesini içerir

Mevcut XHTML çeşitleri şunlardır;

1. XHTML 1.0 Strict: Bu dosya tipi basit sayfalar yaparken kullanılabilir. CSS kullanarak istenilen sonuçları elde etmek kolaylaşır.

2. XHTML 1.0 Transitional: Bu tip ile daha çok HTML'nin avantajları kullanılabilir ve CSS desteği bulunmayan tarayıcılar için de yine bu tip kullanılmalıdır.

3. XHTML 1.0 Frameset : Bu tip HTML'den gelen ve bir pencerede birden fazla sayfayı göstermede kullanılan çerçeve ("frame") yapısını destekler.

XHTML belgesinin well-formed sayılabilmesi için dikkat edilmesi gereken özelliklerin başlıcaları aşağıda verilmiştir. Bu özellikler HTML'de "olmasa da olur" özelliklerdir.

- İyi şekillendirilme ve iç içe doğru yuvalanmış kodlar,
- Küçük harfli kodlar ve özellik değerleri,
- Boş bırakılmayan etiketler,
- Tırnak(" ") içinde yazılacak etiket ve özellik değerleri,
- Kısaltılmamış özellik ve etiket adları,
- Slash (/) ile kapatılmış boş kod veya etiketler,
- Kurallara uygun script ve stil kullanımları.

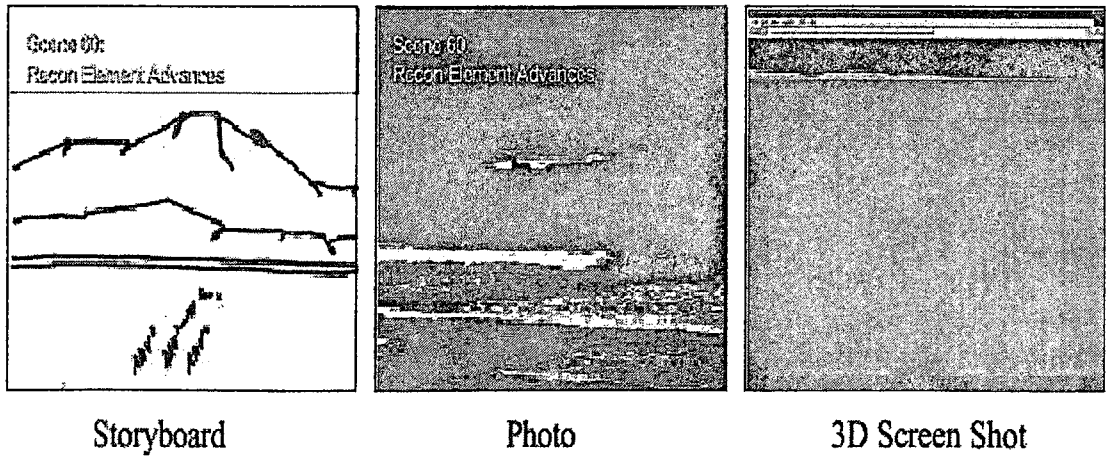
5. BENZER VE İLGİLİ ÇALIŞMALAR

XML ve ilgili teknolojilerin sahip olduğu potansiyeline varmış olan birçok yazılım firması ve bilgi alışverişinin kendileri için vazgeçilmez ve hayati derecede önemi olduğunu bilen yerli yabancı tüm kuruluşlar bu alanda birçok projeyi hayata geçirmeye başlamışlardır.

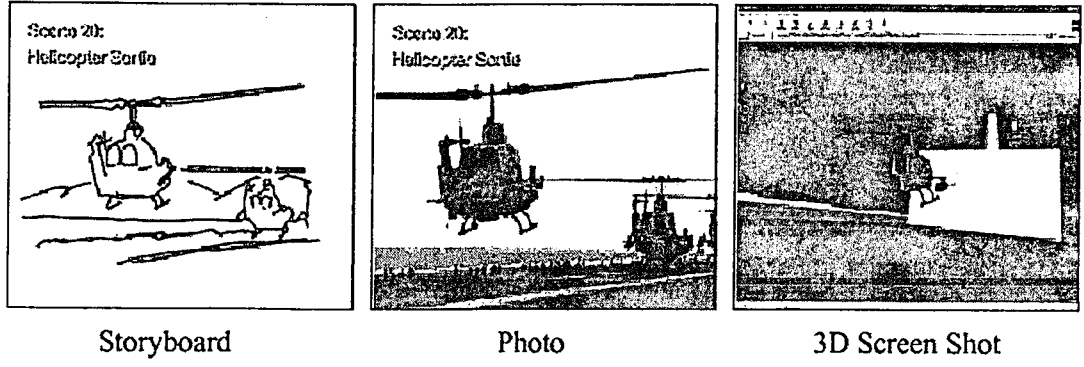
Yapılan bu çalışmanın amacı Türk silahlı Kuvvetlerinin de bu yarışta geride kalmamasına bir nebze katkıda bulunmak olduğundan verilmiş olan örneklerin bazıları yabancı silahlı kuvvetlerde yürütülen proje ve çalışmalardan oluşmaktadır.

5.1. Scenario Authoring And Visualization for Advanced Graphical Environments (Savage)

Proje ABD silahlı kuvvetlerinde hareket emirlerinin XML formatına çevrilmesini ve Web3D, Virtual Reality Modeling Language (VRML), DIS-Java-VRML and Extensible 3D (X3D), GeoVRML teknolojileri ile yine bu XML formatındaki hareket emirlerinin nasıl bir tarayıcı yardımıyla hareket sahasının 3 boyutlu modelleneceğinin üzerinedir.[16,17] Çalışmalar sonucu elde edilmiş görüntülerin bazıları aşağıda verilmiştir.



Şekil 5.1 SAVAGE projesi çıkarma hareketi Web3D görüntüleri



Şekil 5.2 SAVAGE Helikopter hareketi Web3D görüntüleri

5.2. Visualization Of Theater-Level Radio Communications Using A Networked Virtual Environment

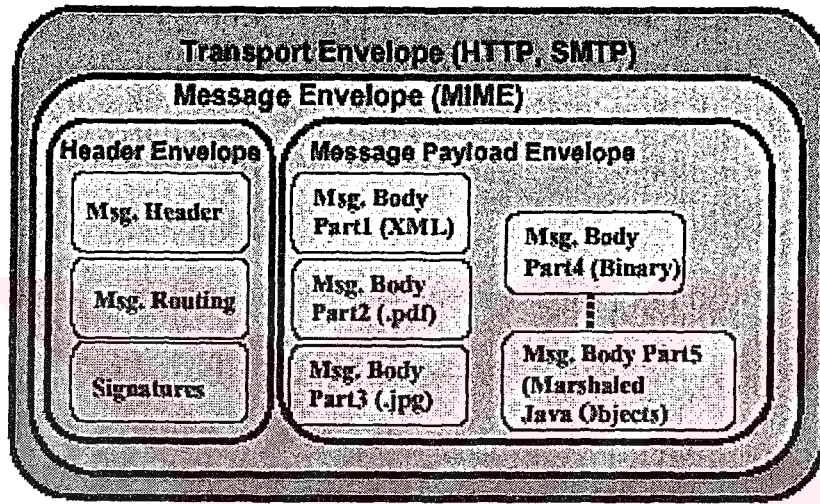
Bu projede ABD silahlı kuvvetlerinde iletişim araçlarının muharebe sahasında konumlarına göre nasıl bir haberleşme ağı kuracağına dair yapılmış, tüm verilerin XML formatına çevrilerek kullanıldığı bir çalışmadır.[18] .Proje çalışmasından elde edilmiş bir görüntü aşağıda verilmiştir.



Şekil 5.3. Harekat Sahasındaki Haberleşme Cihazlarının Yerleşimi ve Kapsama Alanları Görüntüsü

5.3. OASIS ebXML Message Service Protocol

Elektronik ortamda (WEB) veri alışverişi için standart bir protokol oluşturma çabasıyla yürütülen bir projedir. Özellikle ticari faaliyetlerdeki veri alışverişini, XML veri standart ve formatında, http ve RPC protokolüne dayanarak sağlamayı amaçlayan, OASIS (Organization for Advancement of Structured Information Standards) tarafından, açık kaynak kodlu (open-source) yürütülen bir çalışmadır. [19,20] ebXML'in temel yapısı aşağıdaki gibidir.



Şekil 5.4 ebXML'in temel yapısı

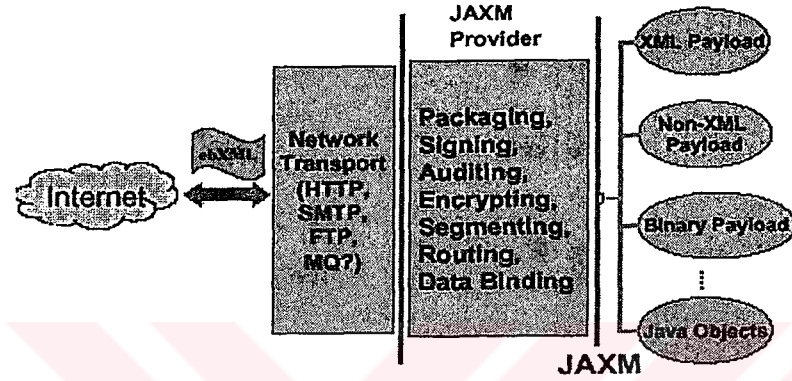
5.4. XML Messaging Protocol (XMLMSG 1.0)

IETF (Internet Engineering Task Force)'in yürüttüğü ve hala internet draft (tamamlanmamış) olan bir çalışmadır. Çalışma aşağıdaki alanlarda standartlar oluşturmayı amaçlamaktadır. [21]

- XML Mesajlaşma için genel elemanlar ve özellikler (XMLMSG-CME),
- XML belgelerinin farklı protokoller üzerinde alışverişi [XMLMSG-DET],
- Güvenilir mesajlaşmanın hatasız olarak yapılması [XMLMSG_RM],
- Şifreli mesajlaşma, veri alışverişinin şifreli olarak yapılması [XMLMSG-SM],
- Veri alışverişinin hangi tipte olacağını kategorize edilmesi XMLMSG-DCD,
- Transfer protokol tanımlamaları (HTTP, SMTP vb.)

5.5 Java™ API for XML Messaging (JAXM)

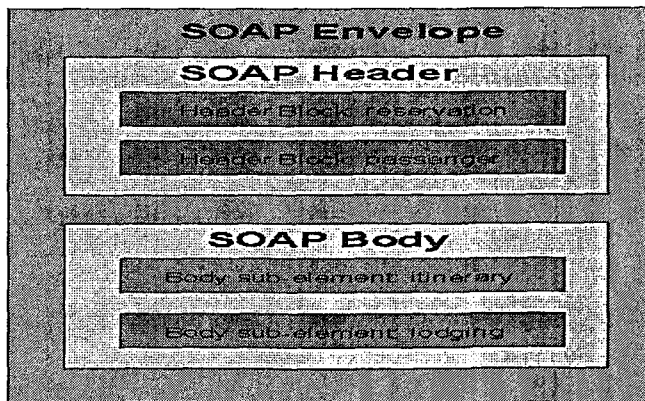
SUN şirketi tarafından Java tabanlı yürütülen XML mesajlaşma için bir API ara yüz oluşturma projesidir. Bu çalışma da açık kaynak kodlu yürütülmektedir. Elektronik ticaret için standartlar daha çok amaç edinilmiştir. SOAP(Simple Object Accessing Protokol)'la çok katmanlı (n-tier) olacak şekilde bir entegrasyon sağlanmıştır. JAXM temel yapısı aşağıdaki gibidir. [22]



Şekil 5.5. JAXM temel yapı modeli

5.6. Simple Object Access Protocol (SOAP)

Dağıtık (Distributed) uygulamalarda ve web servislerinin haberleşmesinde kullanılmak üzere tasarlanan, RPC (Remote Procedure Call) modelini kullanan, istemci/sunucu mantığına dayalı bir protokoldür. Yeni bir teknoloji olmasına karşılık, alt yapısı sayesinde Web Servislerinin adı SOAP la birlikte anılır olmuştur. SOAP mesaj formatı olarak XML, transfer protokolü olarak ta HTTP kullanır. SOAP, HTTP protokolünü kullandığı ve taşıdığı mesajlar düz yazı dosyası halinde olduğu için firewall dostudur. SOAP mesaj formatının temel yapısı Şekil 5.6. da verilmiştir.[23]

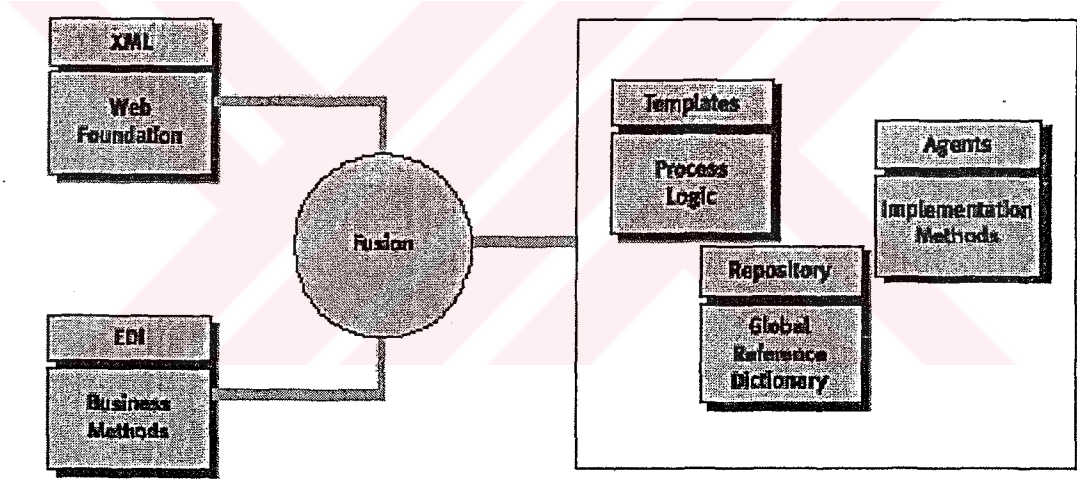


Şekil 5.6. SOAP mesaj formatı

5.7.XML/EDI

EDI, XML framework (iş ortakları arasındaki XML mesaj transferi, iletişim, etkileşim, koordinasyon, için geliştirilmiş kapsamlı bir araçlar ve standartlar kümesi) oluşturmak için yürütülen bir projedir. EDI 1970'lerden itibaren şirketler ve tedarikçileri arasındaki belge değişimini VAN (Value Added Networks) ağlarında başarılı bir şekilde sağlamıştır.

EDI Amerika'da ANSI X.12 ve Avrupa'da EDIFACT mesaj standartlarını kullanmaktadır. Uzun zamandan beri EDI işten B2B işlemlerin gerçekleşmesinde büyük katkısı olan bir teknolojidir. Bununla beraber EDI'nin yüksek maliyeti ve esnek olmayan yapısı bu teknolojinin yaygın olarak kullanılmasını engellemektedir. Yalnızca büyük çaplı firmalar bu teknolojiyi kullanabilmektedir. XML/edi nin temel elemanları aşağıdaki şekildeki gibi gösterilebilir. [24-26]



Şekil 5.7 XML/EDI temel elemanları

6. UYGULAMA

6.1.Geliştirme Ortamı

Geleceğin web uygulamalarında standart olacağı görünen XML'e dayalı, Kara Kuvvetleri Komutanlığı arasında yapılan mesaj alışverişini sağlayan, web tabanlı bir uygulama geliştirmek, bu çalışmanın temelini oluşturmaktadır. Uygulamanın geliştirilmesinde kullanılan donanım ortamı şöyledir.

Öğe	Değer
İşletim Sistemi Adı	Microsoft Windows XP Professional
Sürüm	5.1.2600 Service Pack 1 Yapı 2600
İşletim Sistemi Üreticisi	Microsoft Corporation
Sistem Adı	KKK-KN4AUURS9WK
Sistem Üreticisi	System Manufacturer
Sistem Modeli	System Name
Sistem Türü	X86-based PC
İşlemci	x86 Family 6 Model 4 Stepping 2 AuthenticAMD ~1343 Mhz
BIOS Sürümü/Tarihi	Award Software, Inc. ASUS A7A266 ACPI BIOS Revision 1011, 01.03.2002
SMBIOS Sürümü	2.3
Windows Dizini	C:\WINDOWS
Sistem Dizini	C:\WINDOWS\System32
Önyükleme Aygıtı	\Device\HarddiskVolume1
Yerel Ayar	Türkiye
Donanım Soyutlama Katmanı	Sürüm = "5.1.2600.1106 (xpsp1.020828-1920)"
Kullanıcı Adı	KKK-KN4AUURS9WK\Hayrettin
Saat Dilimi	GTB Standart Saati
Toplam Fiziksel Bellek	512,00 MB'a ulaştığında
Kullanılabilir Fiziksel Bellek	292,08 MB'a ulaştığında
Toplam Sanal Bellek	1,35 GB
Kullanılabilir Sanal Bellek	990,54 MB'a ulaştığında
Disk Belleği Dosyası Alanı	866,57 MB'a ulaştığında
Disk Belleği Dosyası	C:\pagefile.sys

Şekil 6.1 Çalışmanın Gerçekleştirildiği Bilgisayarın Donanım Özellikleri

Tezin geliştirilmesinde kullanılan yazılım teknolojileri aşağıda sıralanmıştır.

1. Active Server Pages(ASP): Microsoft'un geliştirdiği sunucu tarafı dinamik web sayfalarının oluşturulmasında kullanılan programlama dili,[27]
2. JavaScript Programlama Dili: İstemci ve Sunucu tarafı web sayfalarında çalışabilen kodlar yazmayı sağlayan, tarayıcıların nerdeyse tamamının desteklediği dil,[28]
3. Internet Information Services 5.1(ISS): Microsoft'un geliştirdiği ve Windows 2000 Professional ile Windows XP Professional işletim sistemleri ile birlikte gelen kişisel web sunucusu,

4. Microsoft XML CORE SERVICES 4.0 (MSXML): Microsoft'un geliřtirdiđi XML tabanlı uygulamalar geliřtirilmesinde kullanılan DOM ve SAX derleyicileri ile XSL, XMLHTTP ve XPath spesifikasyonlarını barındıran ücretsiz yazılım programı,[14]

5. XML Notepad 1.0: Microsoft'un geliřtirdiđi XML doküman ve belgelerinin oluřturulmasında kullanılan ücretsiz metin editörü,

6. EditPlus 1.2: Program kod yazılımları için kullanılan metin editörü,

7. Cascading Style Sheets (Css 1.0 ve Css 2.0): W3C organizasyonu tarafından kabul edilmiř web sayfalarının görünümlü için kullanılan stil komutları.

6.2.Uygulama için oluřturulan XML dokümanları ve yapıları

Uygulamada oluřturulan XML dokümanları, kullanım amaçları ve yapıları ařađıdaki gibidir.

- Mesaj Indeks Dokümanı (MesajIndex.xml):

Kara Kuvvetlerinde birlikler arasında gönderilen mesajların indekslendiđi xml dokümanıdır. Mesaj metni bu belgede yer almaz. Kullanıcı iřlemlerinin (silme,okuma vb.) gerekleřtirildiđi belgedir. Gönderilen her mesaj için ayrı bir xml dokümanı oluřturulur. MesajIndex.xml dokümanının yapısı(DTD) řu řekildedir.

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<!ELEMENT MesajId (#PCDATA)>
<!ELEMENT Kimden (#PCDATA)>
<!ELEMENT Kime (#PCDATA)>
<!ELEMENT Bilgi (#PCDATA)>
<!ELEMENT OBilgi (#PCDATA)>
<!ELEMENT OKime (#PCDATA)>
<!ELEMENT Gizlilik (#PCDATA)>
<!ELEMENT Konu (#PCDATA)>
<!ELEMENT Dosya (#PCDATA)>
<!ELEMENT AISil (#PCDATA)>
<!ELEMENT GonSil (#PCDATA)>
<!ELEMENT BilgiSil (#PCDATA)>
<!ELEMENT AIOku (#PCDATA)>
<!ELEMENT BilgiOku (#PCDATA)>
<!ELEMENT Mesaj (MesajId, Kimden, Kime, Bilgi, OKime, OBilgi, Gizlilik, Konu
,Dosya,AISil, GonSil, BilgiSil, AIOku, BilgiOku)>
<!ELEMENT MesajIndex (Mesaj+)>
```

- Mesaj Dokümanları(MesajId.xml):

Mesajla ilgili tüm bilgilerin tutulduğu esas xml dokümanlarıdır. Her mesaj için MesajIndex.xml dokümanında belirtilen MesajId'ye uygun bir xml dokümanı yaratılmaktadır. Böylece parser'ın (işleyici) tüm mesajları hafızaya alarak hafızanın doldurulması ve kullanıcıyı ilgilendirmeyen mesajlarla vakit harcanmaması sağlanmaktadır. Mesajların metin alanları belgelere şifreli(kriptolu) olarak kaydedilmektedir. Gönderilen her mesaj için oluşturulan xml dokümanlarının yapısı(DTD) şu şekildedir.

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<!ELEMENT Bilgi (#PCDATA)>
<!ELEMENT Dosya (#PCDATA)>
<!ELEMENT Ek (#PCDATA)>
<!ELEMENT Gizlilik (#PCDATA)>
<!ELEMENT Haber (#PCDATA)>
<!ELEMENT Imza (#PCDATA)>
<!ELEMENT Kgk (#PCDATA)>
<!ELEMENT Kimden (#PCDATA)>
<!ELEMENT Kime (#PCDATA)>
<!ELEMENT Konu (#PCDATA)>
<!ELEMENT Koordinasyon (#PCDATA)>
<!ELEMENT MesajId (#PCDATA)>
<!ELEMENT MesajNo (#PCDATA)>
<!ELEMENT Metin (#PCDATA)>
<!ELEMENT OBilgi (#PCDATA)>
<!ELEMENT OKime (#PCDATA)>
<!ELEMENT Referans EMPTY>
<!ELEMENT Tsg (#PCDATA)>
<!ELEMENT Mesaj (MesajId, Kimden, Kime, Bilgi, OKime, OBilgi, Gizlilik,
Konu, Tsg, Ek, Kgk, MesajNo, Referans, Koordinasyon, Imza, Haber, Dosya, Metin)
>
```

- Online Hat Dokümanı(sohbet.xml) :

Kullanıcıların karşılıklı olarak mesajlaştıkları online hat yazışmalarının tutulduğu xml dokümanıdır. Bu dokümanda daima son on yazışma yer almaktadır. Online hat dokümanının yapısı(DTD) şu şekildedir.

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<!ELEMENT kimden (#PCDATA)>
<!ELEMENT metin (#PCDATA)>
<!ELEMENT renk (#PCDATA)>
<!ELEMENT mesaj (kimden,metin,renk)>
<!ELEMENT mesajlar (mesaj+)>
```

- Üye Bilgileri Dokümanı(Üye.xml):

Kullanıcıların(üyelerin), bilgilerinin yer aldığı xml dokümanıdır. Uygulama girişinde üyelik kontrolü de bu belgedeki verilerle yapılmaktadır. Dokümanda şifre ve hatırlatıcı soru alanları şifreli olarak kaydedilmektedir. Üye bilgileri dokümanının yapısı(DTD) şu şekildedir.

```
<?xml version="1.0" encoding=" ISO-8859-9"?>
<!ELEMENT Uyeler (uye+)>
<!ELEMENT uye EMPTY>
<!ATTLIST uye
    EMSNo CDATA #REQUIRED
    ad CDATA #REQUIRED
    sinif CDATA #REQUIRED
    rutbe CDATA #REQUIRED
    sicil CDATA #REQUIRED
    gorev CDATA #REQUIRED
    gyer CDATA #REQUIRED
    sifre CDATA #REQUIRED
    soru CDATA #REQUIRED
    cevap CDATA #REQUIRED
    aktif CDATA #REQUIRED
    onay CDATA #REQUIRED
    yetki CDATA #IMPLIED
    tarih CDATA #IMPLIED
>
```

- Yardımcı dokümanlar (Öncelik.xml,Gizlilik.xml,Rütbe.xml,Sınıf.xml) :

Kullanıcıların rütbeleri, sınıfları ile mesajların öncelik, gizlilik derecelerinin tespiti için oluşturulmuş xml dokümanlarıdır. Bu dokümanlardan elde edilen veriler uygulamada seçmeli kutularda(combobox) kullanılmaktadır. Böylece kullanıcıya alanlara girmesi gereken veriler hazır olarak sunulmaktadır. Bu xml dokümanlarının yapıları (DTD) genel olarak birbirine benzemektedir. Burada örnek açısından sadece gizlilik.xml dokümanının yapısı verilmiştir.

```
<?xml version="1.0" encoding=" ISO-8859-9"?>
<!ELEMENT ROOT (gizli+)>
<!ELEMENT gizli (Derece)>
<!ELEMENT Derece (#PCDATA)>
```

6.3.Uygulama Modülleri

6.3.1.Giriş Ekranı Modülü

Kullanıcı girişinin yapıldığı ve sistemdeki aktif(online) kullanıcı sayısının belirlendiği sistemin karşılama sayfasıdır.Şekil 6.2.

Merkeze Bağlanmak İçin Kullanıcı Adı ve Şifrenizi Girmiz

Kullanıcı Adı: Şifre :

Kullanıcı Adınız EMS Numaranızdır Şifreniz 4-8 Karakter Uzunluğundadır

[Yeni Kullanıcı](#) [Sistem Hakkında](#)

Kullanıcı Adı ve Şifre Doğrulanamadı!Küçük Büyük Harf duyarlılığına Dikkat Edin
Sistemde (0) Kullanıcı Aktif

©Telif Hakkı Kara Kuvvetleri Komutanlığı 2003

Şekil 6.2 Giriş Ekranı Modülü

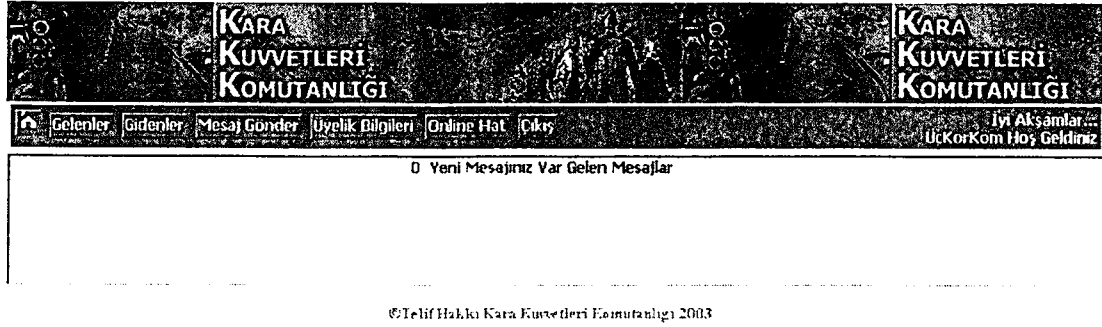
İki JavaScript fonksiyonu kullanılmıştır.

function aktif_uye_say(): Bu fonksiyonla aktif üye sayısı tespit edilmektedir.

function girisform_onsubmit(): Bu fonksiyonla kullanıcının formun kullanıcı adı ve şifre alanlarına yanlış giriş yapması engellenmekte ve kullanıcı ikaz edilmektedir.

Forma girilen veriler doğru ise XMLHTTP protokolü[28,29] ile kullanıcının girdiği veriler xml formatında sunucuya, uyekontrol.asp dosyasına gönderilmektedir. Bu dosyada kullanıcı veri girişlerinin doğruluk kontrolü yapılmaktadır.Veriler yanlış ise cevap giriş sayfasına(giris.asp) gönderilmektedir. Gönderilen cevap giris.asp sayfasındaki ikaz katmanında(div) gösterilmektedir. Bu işlemler yapılırken sayfa değişmemektedir(yenilenmemektedir). Bu XMLHTTP' nün bize sağladığı yeniliklerden biridir. Veriler doğru ise kullanıcı verileri session ve çerez(cookies) değişkenlerine kaydedilerek mesajmrk.asp dosyasına yani mesaj merkezine yönlendirilmektedir.

6.3.2. Mesaj Merkezi Modülü



Şekil 6.3 Mesaj Merkezi Karşılama Modülü Ekranı

Kullanıcının sisteme bağlandıktan sonraki karşılama ekranıdır. Şekil 6.3. Kullanıcıya henüz okunmamış yeni gelen mesaj sayısı bildirilmektedir. Kullanıcı tüm mesaj işlemlerini bu modülde gerçekleştirmektedir. JavaScript fonksiyonu, function mesajmrk() ile kullanıcının okumadığı yeni gelen mesaj sayısı, giden, gelen mesajlar xml dokümanlarından belirlenmektedir. Üst menüdeki linklere tıklanarak ilgili işlemler gerçekleştirilmektedir. İşlem sonuçları main adlı iframe'de gösterilmektedir. Bu modül 6 alt modüle sahiptir. Bunlar;

1. Gelen Mesaj Kutusu Modülü,
2. Giden Mesaj Kutusu Modülü,
3. Mesaj Gönderme Modülü,
4. Üyelik Bilgileri Modülü,
5. Online Hat Modülü,
6. Çıkış Modülü' dür.

6.3.2.1. Gelen Mesaj Kutusu Modülü

Sil	Kimden	Konu	Öncelik	Gizlilik	Dosya
X1	KKKKom	SAYTEK Kongresi	Çok İvedi	Gizli	-
X1	KKKKom	Yönerge Değişikliği	Normal	Hizmete Özel	1
X1	KKKKom	Kamp Başvuruları	İvedi	Özel	-
X1	KKKKom	Akademi Hazırlık Dokümanları	Normal	Hizmete Özel	1
X1	KKKKom	Yüksek Lisans Programına Seçilen Personel	İvedi	Hizmete Özel	1

©Telif Hakkı Kara Kuvvetleri Komutanlığı 2003

Şekil 6.4. Gelen Mesaj Kutusu Modülü Ekranı

2.function msgsil(mesajid,kod) : İstenilen mesajı silmek için kullanılır. Msgsys.asp dosyasına mesajid ile ulaşır ve mesajın MesajIndex.xml dokümanındaki silindi bilgisini değiştirir. Dikkat edilirse mesaj dokümanları aslında silinmemektedir. ki bu hatalı silinmelerden kaynaklanacak problemlere mani olmaktadır.

Şekil 6.5. Mesaj Okuma Ekranı

6.3.2.2. Giden Mesaj Kutusu Modülü

KARA KUVVETLERİ KOMUTANLIĞI					
Gelenler	Gidenler	Mesaj Gönder	Üyelik Bilgileri	Online Hat	Dosya
Giden Mesaj Kutusu					
Sil	Kime	Konu	Öncelik	Gizlilik	Dosya
X1	ÜçKorkKom	SAYTEK Kongresi	Çok İvedi	Gizli	-
X1	ÜçKorkKom	Yönerge Değişikliği	Normal	Hizmete Özel	0
X1	ÜçKorkKom	Kamp Başvuruları	İvedi	Özel	-
X1	ÜçKorkKom	Akademik Hazırlık Dokümanları	Normal	Hizmete Özel	0
X1	ÜçKorkKom	Yüksek Lisans Programına Seçilen Personel	İvedi	Hizmete Özel	0

©Telif Hakkı Kara Kuvvetleri Komutanlığı 2003

Şekil 6.6. Giden Mesaj Kutusu Modülü Ekranı

Kullanıcının göndermiş olduğu mesajların gösterildiği modüldür. Şekil 6.6. Kullanıcı mesajın göndermiş olduğu kişi tarafından okunup okunmadığını bu modülde görebilir. Mesaj alıcı tarafından okunmuşsa açık zarf resmiyle, okunmamışsa ok resmiyle gösterilmektedir. İsterse göndermiş olduğu mesajları görebilir yada giden mesaj kutusundan silebilir. Burada da gelen kutusu modülündeki fonksiyonlar kullanılmaktadır.

6.3.2.3. Mesaj Gönderme Modülü

Gönderilecek mesajların hazırlandığı modüldür. Şekil 6.7. Mesaj, üst bilgiler, mesaj metni ve mesaj alt bilgiler olmak üzere üç bölümden oluşmaktadır. Form ilgili bilgiler yardımcı xml dokümanlarından alınmaktadır. Form giriş kontrolleri yapılmakta ve eksik yada hatalı mesaj gönderimi engellenmektedir. Mesaj Metni ayrı bir modülde hazırlanmakta ve gönderim anında forma dahil edilmektedir.

Gönderilecek mesaj MD5 dijital imzalama tekniğiyle imzalanarak mesajın alıcıya değişime uğramadan gidip gitmediğinin kontrolü yapılmaktadır. Ayrıca mesaj, göndericinin ve alıcının anahtarları ile kriptolanmaktadır. Kullanıcıya, mesajın alıcıya doğru bir şekilde ulaşmış olup olmadığı ikazlar ile geri bildirilmektedir.

Mesaja dosya ekleme işlemi de bu modülde gerçekleştirilmektedir. Dosyaların eklenmesi XMLHTTP protokolüyle gerçekleştirilmektedir. Sadece bir dosya eklemeye müsaade edilmektedir.

KARA KUUVETLERİ KOMUTANLIĞI

KARA KUUVETLERİ KOMUTANLIĞI

Gelenler Gidenler Mesaj Gönder Üyelik Bilgileri Online Hat Çıkış İyi Akşamlar! UçKorKom Hız Geldiniz

Mesaj Gönder

Dosya Ekle C:\Documents and Settings\... Gözet Dosya Gönder

Dosya Eklendi

Öncelik Derecesi Gereği İçin Normal Bilgi İçin Normal

Tarih Saat Grubu TSG Ali

Haber Talimatı

Kimden ÜçKorKom Ön Ek

Kime ÜçKorKom Gizlilik Derecesi Hizmete Özel

Bilgi ÜçKorKom Mesaj No

KGK

Konu

Mesaj Metni

1. Yüksek lisans yapmakta olan personelin isimleri Ek te gönderilmiştir.

- Personelle ilgili bilgiler Ek B deki formata uygun olarak doldurulup gönderilecektir.

Rica ederim

Mesaj Metnini Kaydet

Referans Yapılan Mesaj Koordinasyon Müsaade Eden

Hayrettin DÖNMEZ Hayrettin DÖNMEZ

Temizle Gönder

Şekil 6.7. Mesaj Gönderme Modülü Ekranı

Bu modül içinde kullanılan JavaScript fonksiyonları şunlardır.

1. function Tarih(date,format) : Kara Kuvvetlerinde kullanılan tarih saat formatını oluşturmaktadır. Örneğin mesaj 07 Haziran 2002 tarihinde saat 1530 da gönderilmiş ise bu xml dokümanına 071530 C HAZ02 şeklinde kaydedilecektir.

2. function kontrollet(): Formun alanlarına doğru veri girişini sağlamaktadır. Yanlış girişlerde kullanıcı ikaz edilmektedir.

3. function MD5(sMessage): Mesaj metnin MD5 dijital imzalama tekniğiyle imzalanmasını sağlayan fonksiyondur. Sunucuya ulaşan mesaj bu teknikle bir daha imzalanmakta ve gönderilen ve alınan mesaj imzaları karşılaştırılarak mesajın doğruluk kontrolü yapılmaktadır. Eğer mesaj hatalı ulaşmışsa alıcıya yansıtılmamakta ve gönderene mesajın ulaşmadığına dair ikaz gönderilmektedir.

4. function KriptoOn/Off(text1,text2): Gönderen ve alıcının anahtarları alınarak mesajın şifrelenmesini yani kriptolanmasını sağlayan fonksiyondur.

5. function dosyaekle(fileName): Dosya ekleme işlemini gerçekleştirir.

6.3.2.4. Üyelik Bilgileri Modülü

Üyelik Bilgileriniz			
Kullanıcı Adı	76134065	Ad ve Soyad	Hayrettin DÖNMEZ
Şifre	Mıhıhare	Rütbe	Üsteğmen
Sicil No	1997-01	Görevi	İk.K.
Görev Yeri	ÜçKorKom	Şifre	5555
Hatırlatıcı Soru	en sevdiğim renk	Soru Cevabı	kesinlikle mavi

Şekil 6.8. Üyelik Bilgileri Modülü Ekranı

Bu modülde msgmrksys.asp dosyasına bağlanılarak, kullanıcı id'sine göre üye.xml dokümanından kullanıcının bilgileri çekilmekte ve ekrana yansıtılmaktadır.Şekil 6.8. İşlem function uyebak(), Javascript fonksiyonu ile gerçekleştirilmektedir.

6.3.2.5. Online Hat Modülü

Kullanıcıların online olarak karşılıklı görüşmesini sağlayan modüldür.Şekil 6.9. Yazışmalar sohbet.xml dokümanında tutulmaktadır. Daima en son on yazışma ekranda gösterilmekte diğer yazışmalar silinmektedir. Yazışmaların gönderilip alınması xml formatındadır.

!!!Online Hattta Gizlilik Derecesi Olan Yazışmalar Yapmamaya Dikkat Ediniz!!!

ÜçKorKom:> Tamam ☺

ÜçKorKom:> Hemen Yolluyorum

KKKKom:> İyi bende bir problem var sandım

ÜçKorKom:> No problema

KKKKom:> Selam

KKKKom:> mesaj ulaştı mı?

ÜçKorKom:> aldım tamam

ÜçKorKom:> gereği yapmakta

Kullanıcı Adı: ÜçKorKom

Mesaj: [Gönder]

☐ Siyah ☐ Mavi ☐ Kırmızı ☐ Yeşil

Şekil 6.9. Online Hat Modülü Ekranı

6.3.2.6. Çıkış Modülü

Kullanıcı sistemi terk ettiğine dair ikaz edilmekte, ikaza verilen cevap evet ise üyeterket.asp dosyasına bağlanarak artık kullanıcının aktif olmadığı bilgisinin üye.xml dokümanına işlenmesi sağlanmaktadır. Böylece kullanıcının sistemi terk etmesi gerçekleşmiş olmaktadır.

6.3.3.Yeni Üye Başvuru Modülü

The screenshot shows the 'Yeni Üye Kaydı' (New Member Registration) form. The form is titled 'Yeni Üye Kaydı' and is part of the 'KARA KUVVETLERİ KOMUTANLIĞI' system. The form includes the following fields:

- Kullanıcı Adı (Username): [Text Field]
- Ad ve Soyad (Name and Surname): [Text Field]
- Sınıf (Class): [Dropdown Menu, currently showing 'Piyade']
- Rütbe (Rank): [Dropdown Menu, currently showing 'Teğmen']
- Sicil No (Service Number): [Text Field]
- Görev Yeri (Duty Station): [Text Field]
- Şifre (Password): [Text Field]
- Şifre Tekrar (Password Repeat): [Text Field]
- Hatırlatıcı Soru (Reminder Question): [Text Field]
- Soru Cevabı (Question Answer): [Text Field]

At the bottom of the form, there are two buttons: 'Temitle' (Cancel) and 'Gönder' (Send). Below the form, there is a link labeled 'Ana Sayfa' (Home Page).

©TeflFlakki Kara Kuvvetleri Komutanlığı 2003

Şekil 6.10. Üyelik Başvuru Modülü Ekranı

Bu modül yeni üye başvuruları için hazırlanmıştır.(Şekil 6.10) Bölümde form kontrolleri mevcuttur. Üye başvuru bilgileri üyeler.asp dosyasına gönderilmekte, bu dosyada yapılan kontrollerle üyenin var olan bir üyenin kullanıcı adıyla müracaat edip etmediği anlaşılmakta, başvuran ikaz edilmektedir. Üyelik başvurusunda bulunanlar ancak sisteme üyeliği onaylandıktan sonra girebilmektedirler.

6.3.4.Sistem Bilgileri Modülü

Sistem hakkında özet bilginin bulunduğu modüldür. Kullanıcıların sistemi kullanmak için sahip olması gereken yazılımlar belirtilmekte ve bu bölümden bu yazılımların indirebilmeleri sağlanmaktadır.

7. SONUÇLAR VE ÖNERİLER

XML, elektronik ticaret, elektronik veri değişimi, tedarik zinciri bütünleştirmesi, iş akışı (workflow), veri yönetimi, akıllı arama makineleri gibi bir çok alanda stratejik bir araç olarak kullanılmaya başlanmış, basit ve esnek metin biçimi teknolojisidir. XML'in en fazla ilgi çeken tarafı web uygulamalarında evrensel bir veri değişim formatı olarak kullanılmasıdır. Bugünkü bilgisayar iletişimindeki çok büyük gelişimlere ve birikime rağmen, bilgisayar sistemlerindeki ve veritabanlarındaki farklı formatlardaki verilerin şirket içi ve şirketler arası taşınması ve işlenmesi en büyük problemlerden birisidir. XML'in özellikleri veri yapılarını, içeriklerini ve kavramlarını platform, şirket ve dilden bağımsız bir yapıda temsiline imkan vermektedir. [26]

Kara Kuvvetleri mesaj sisteminin xml tabanlı hale getirmek bize XML'in sağladığı tüm avantajları vermiş olacağı aşikardır. Bu tez çalışması bu amaca yönelik atılmış olan bir başlangıç adımıdır. Bu adımların daha büyük ve daha hızlı atılması gerekliliği vardır. XMLHTTP protokolü, daha hızlı ve firewalllardan etkilenmemesi nedeniyle tercih edilmiştir. Tez çalışmasında;

- 1.KKK mesaj formatı xml dokümanı haline getirilmiştir,
- 2.Veri değişim formatı olarak xml kullanılmıştır,
- 3.Xml verinin alışverişi XMLHTTP protokolünde yapılmıştır,
- 4.İstemci ile sunucu arasındaki veri akışı kriptolu(şifreli) olarak sağlanmıştır,
- 5.İşlemlerin çoğunluğu JavaScript kodları ile istemci tarafında gerçekleştirilmiştir,
- 6.Sisteme girişler üyelik sistemiyle sağlanmıştır,
- 7.Gönderilen ve alınan mesajlar MD5 dijital imzalama tekniğiyle imzalanmıştır,
- 8.Cascading Style Sheets(Css) görüntüleme teknikleri kullanılmıştır.

Bu alıřma sonucunda, KKK'de kullanılan mesaj formatı XML'e dnřtrldğnden veriler platform, řirket ve dilden bağımsız hale getirilmiřtir. Bylelikle veriler, kolaylıkla tařınabilecek ve istenilen platformda kolaylıkla kullanılabilir. Mesaj ierikleri, XML dokmanlarında tutulduğndan herhangi bir veritabanı uygulamasına da gereksinim kalmamıřtır.

Bu web uygulaması daha da zenginleřtirilebilir. rneğın uygulamaya onay bekleyen ye bařvurularını onaylayacak ve silindi bilgisine sahip xml dokmanlarını sistemden silecek bir ynetici blm eklenebilir.



KAYNAKLAR

- [1] **Akyokuş, S.**, 2002. XML ve XML Uygulamalarına Giriş, PcNet Dergisi Eki, İstanbul.
- [2] A Brief History of the Development of SGML,
<http://www.sgmlsource.com/history/sgmlhist.htm>
- [3] **Marchal, B.**, 2000. XML By Example. Que Pres, USA.
- [4] Extensible Markup Language (XML), <http://www.w3c.org/XML>.
- [5] XML Kütüphanesi, http://www.softwareag.com/turkiye/XML_Kutuphanesi/
- [6] XML Eğitimi, http://www.geocities.com/resit_galip/egitimana.htm.
- [7] XML Eğitimi, <http://hosting.msugs.ch/101nokta/xml-tr0.htm>.
- [8] XML Nedir?, <http://internethaftasi.sdu.edu.tr/XML/HTML/html.html>.
- [9] **Ray, T., E.**, 2001. Learning XML, Oreilly Publishing, USA.
- [10] **Schwartzbach, I. M.**, 2001. XML Mini-Tutorial,
<http://www.brics.dk/~mis/ITU/XML/>
- [11] **Meggison, D.**, 1998. Structuring XML Documents, Prentice Hall Computer Books, USA.
- [12] **Morrison, M.**, 1999. XML Unleashed, Sams Publishing, Indiana, USA.
- [13] **Anderson, R.**, 2000. Professional XML, Wrox Press Ltd, New York.
- [14] <http://msdn.microsoft.com/xml/default.asp>
- [15] Building Documents with XML, XSL, and CSS
<http://www.insidedhtml.com/tips/xml/ts01/page1.asp>
- [16] **Nicklaus, S. D.**, 2001. Scenario Authoring and Visualization For Advanced Graphical Environments (SAVAGE), *Yüksek Lisans Tezi*, Naval Postgraduate School, California.
- [17] **Murray, M. W. and Quigley, J. M.**, 2001. Automatically Generating A Distributed 3d Battlespace Using Usmtf and Xml-mtfair Tasking Order, *Yüksek Lisans Tezi*, Naval Postgraduate School, California.
- [18] **Laflam, W. D.**, 2000. 3d Visualization of Theater-Level Radio Communications Using a Networked Virtual Environment, *Yüksek Lisans Tezi*, Naval Postgraduate School, California.
- [19] <http://www.oasis-open.org/cover/>
- [20] <http://ebxml.org>
- [21] XML Messaging (IETF), <http://xml.coverpages.org/xmlMessagingIETF.html>.
- [22] <http://www.sun.com/software/xml/>
- [23] Simple Object Access Protocol (SOAP), <http://www.w3.org/TR/SOAP/>

- [24] **Güven, M.**, 2002. Integrating a Data Exchange System Based on EDI and E-mail to a Client/Server Application, *Yüksek Lisans Tezi*, Dokuz Eylül University, İzmir.
- [25] **Güven, M.**, 2002. An Electronic Data Iterchange (EDI) Implemantation Via E-mail in Law Automation System, *Yüksek Lisans Tezi*, Dokuz Eylül University, İzmir.
- [26] **Akyokuş, S. ve KİLİMCİ P.**, 2002. Kurumlar Arası (B2b) Ticaret Ve Biztalk 2000server İle Bir B2b Uygulaması, Doğuş Üniversitesi, İstanbul.
- [27] **Wallther, S., and Levine, J.**, 2001. ASP ile E-Ticaret Programcılığı, Sistem Yayıncılık, İstanbul
- [28] **Pillai, P.**, 2002. XML and JavaScript Tutorial, http://www.codetoad.com/xml_javascripti_tutorial.asp.
- [29] **Harper, M.**, 2002. The Power of the XMLHTTP Library, <http://www.devarticles.com/art/1/25/2>.
- [30] **Dumbill, E.**, 2000. XML/HTTP Messaging: Good, Getting Beter, <http://www.xml.com/pub/a/2000/06/28/devcon/messaging.htm>.
- [31] Hypertext Transport Protocol (HTTP), <http://www.w3.org/Protocols/rfc2616/rfc2616.html>.

EKLER

EK-A: Uygulamanın kaynak kodları



