

**ANLAMSAL WEB TEKNİKLERİ KULLANARAK GPS TABANLI
BAĞLAM BİLİNÇLİ MOBİL UYGULAMA**

Olgun KARADEMİRCİ

**YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

ŞUBAT 2008

ANKARA

Olgun KARADEMİRCİ tarafından hazırlanan ANLAMSAL WEB TEKNİKLERİ KULLANARAK GPS TABANLI BAĞLAM BİLİNÇLİ MOBİL UYGULAMA adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Doç. Dr. M. Ali AKCAYOL
Tez Danışmanı, Bilgisayar Mühendisliği

Bu çalışma, jürimiz tarafından oy birliği ile Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir.

Prof. Dr. M. Sezai DİNÇER
Elektrik Elektronik Mühendisliği, Gazi Üniversitesi

Doç. Dr. M. Ali AKCAYOL
Bilgisayar Mühendisliği, Gazi Üniversitesi

Prof. Dr. İskender ÖKSÜZ
Kimya Mühendisliği, Gazi Üniversitesi

Tarih: 14/02/2008

Bu tez ile G.Ü. Fen Bilimleri Enstitüsü Yönetim Kurulu Yüksek Lisans derecesini onamıştır.

Prof. Dr. Nermin ERTAN
Fen Bilimleri Enstitüsü Müdürü

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü fikir ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

Olgun KARADEMİRÇİ

**ANLAMSAL WEB TEKNİKLERİ KULLANARAK GPS TABANLI BAĞLAM
BİLİNÇLİ MOBİL UYGULAMA
(Yüksek Lisans Tezi)**

Olgun KARADEMİRCİ

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
Şubat 2008**

ÖZET

Bu çalışmada anlamsal web servisleri kullanılarak GPS tabanlı bağlam bilinçli mobil bir uygulama geliştirilmiştir. Mobil uygulama, anlamsal web yöntemleri ile modellenen verileri barındıran merkezi sistem ile haberleşmektedir. Böylece kullanıcıya ait profil bilgileri ve çevresel bilgiler merkezi sisteme iletilmektedir. Merkezi sistemde bulunan veriler ile kullanıcıdan gelen veriler anlamsal web sorgulama tekniklerine göre işlenmekte ve ortaya çıkan bağlam bilinçli bilgiler mobil istemciye gönderilmektedir.

Geliştirilen uygulama ile klasik yöntemlerle oldukça uzun zaman harcanarak elde edilebilen bilgilerin mobil cihaz üzerinde bağlam bilinçli yöntemler kullanılarak otomatik olarak üretilmesi ve kullanıcıya sunulması sağlanmıştır. Böylece bu uygulama ile kullanıcılara yönelik kişiselleşmiş bilgilere ulaşım hızlanmış ve maliyetler düşmüştür. Ayrıca bilgilerin ortak kullanımına yönelik bilgi sınıflandırılması sağlanmıştır.

Bilim Kodu : 902.1.014
Anahtar Kelimeler : Anlamsal web, mobil servisler, bağlam bilinçli servisler
Sayfa Adedi :102
Tez Yöneticisi : Doç. Dr. M. Ali AKCAYOL

**GPS BASED CONTEXT-AWARE MOBILE APPLICATION USING
SEMANTIC WEB TECHNIQUES**

(M.Sc. Thesis)

Olgun KARADEMİRÇİ

**GAZİ UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY**

February 2008

ABSTRACT

In this study, using semantic web techniques a GPS based, context-aware mobile application has been developed. The mobile application communicates with a central system that hosts the data modeled by the semantic web techniques. Therefore, user information and information on the environment are sent to the central system. The data stored in the central system and the data received from the user are processed using the semantic web query techniques, and context-aware information thus created by these techniques is sent to the mobile client.

With the present application, information that could only be obtained through long and retrospective work through traditional methods is created automatically using context-aware methods and served to the users. Through the use of the present application, personalized information can be obtained in a shorter time at a reduced cost. Further, the data classification for facilitating sharing of the data is obtained.

Science Code : 902.1.014
Key Words : Semantic web, mobile services, context-aware services
Page Number : 102
Adviser : Assoc. Prof. Dr. M. Ali AKCAYOL

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren hocam Doç. Dr. M. Ali AKCAYOL'a yine kıymetli tecrübelerinden faydalandığım hocalarım Doç. Dr. Şeref SAĞIROĞLU ve Yrd. Doç. Dr. Hasan Şakir BİLGE'ye teşekkürü bir borç bilirim.

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT	v
TEŞEKKÜR	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ	ix
ŞEKİLLERİN LİSTESİ	x
RESİMLERİN LİSTESİ	xii
SİMGELER VE KISALTMALAR	xiii
1. GİRİŞ	1
2. ANLAMSAL WEB	8
2.1. Web	9
2.1.1. Statik web	10
2.1.2. Dinamik web	12
2.2. Web Servisleri	13
2.2.1. Web servis teknolojileri	14
2.2.2. Web servis teknolojilerinin uygulanması	23
2.3. Anlamsal Web	24
2.3.1. Anlamsal web ve geleneksel web karşılaştırması	26
2.3.2. Anlamsal web mimarisi	28
2.3.3. Anlamsal web uygulama örnekleri	35

Sayfa

2.3.4. Anlamsal web yazılım araçları.....	37
3. BAĞLAM BİLİNÇLİ UYGULAMALAR	42
3.1. İçerik ve Bağlam.....	43
3.2. Bağlam Farkındalığı.....	45
3.3. Mobil Cihazlarda Bağlam Bilinçli Uygulamalar	46
4. ANLAMSAL WEB TEKNİKLERİ KULLANARAK GPS TABANLI BAĞLAM BİLİNÇLİ MOBİL UYGULAMA.....	49
4.1. Sistem Mimarisi.....	51
4.1.1. İstemci mimarisi.....	52
4.1.2. Sunucu mimarisi.....	56
4.2. Mobil İstemci Modülü.....	57
4.3. Web İstemci Modülü.....	69
4.4. Sunucu Uygulama Modülü.....	72
4.5. Anlamsal ve Geleneksel Veri Yönetim Modülü.....	75
5. SONUÇ VE ÖNERİLER	79
KAYNAKLAR.....	81
EKLER.....	86
EK-1 Maris sinema veri modeli RDF şema	87
EK-2 Maris sinema veri modeli RDF	93
ÖZGEÇMİŞ.....	102

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. Web teknolojileri evrimi.....	8
Çizelge 2.2. Geleneksel ve anlamsal web uygulamalarının karşılaştırılması	28
Çizelge 3.1. En çok ziyaret edilen mobil web sayfaları (Haziran 2006)	48

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. Web sayfalarının artış grafiği	9
Şekil 2.2. Web veri modeli.....	11
Şekil 2.3. Web servis temel modeli	15
Şekil 2.4. Servis istemci ve servis sağlayıcı süreçleri.....	17
Şekil 2.5. XML işaretleme örneği	18
Şekil 2.6. SOAP istek mesajı örneği.....	20
Şekil 2.7. SOAP cevap mesajı örneği	21
Şekil 2.8. Standart web işleyişi.....	27
Şekil 2.9. Anlamsal web işleyişi.....	27
Şekil 2.10. Anlamsal web mimari modeli.....	29
Şekil 2.11. RDF deyimi grafik gösterimi.....	31
Şekil 2.12. RDF deyimi grafik gösterim örneği	32
Şekil 2.13. RDF çoklu deyim grafik gösterim örneği.....	32
Şekil 2.14. RDF deyimi XML tabanlı gösterimi	32
Şekil 2.15. RDFS için sınıf hiyerarşisi örneği.....	33
Şekil 2.16. Oracle, anlamsal teknolojiler merkezi.....	36
Şekil 2.17. Adobe, extensible metadata platform.....	36
Şekil 2.18. Hakia, anlamsal arama motoru	37
Şekil 2.19. Sesame mimari yapısı	39
Şekil 2.20. Sesame sorgu çözümleme ve optimize etme modeli.....	39

Şekil	Sayfa
Şekil 2.21. Sesame yapılandırma arabirimi.....	40
Şekil 2.22. Protege kullanıcı arabirimi	41
Şekil 3.1. Geleneksel bilgisayar uygulamaları	42
Şekil 3.2. İçerik bilinçli uygulamalar.....	43
Şekil 3.3. Uygulama ve kullanıcı açısından bilgi erişimi karşılaştırması.....	45
Şekil 4.1. MARIS platformu temel modülleri.....	50
Şekil 4.2. MARIS genel mimari	51
Şekil 4.3. MARIS mobil istemci mimarisi.....	54
Şekil 4.4. MARIS web istemci mimarisi	55
Şekil 4.5. MARIS kurumlar web servisi mimarisi	55
Şekil 4.6. MARIS sunucu mimarisi	56
Şekil 4.7. Eclipse IDE	57
Şekil 4.8. MARIS, web yönetici arabirimi giriş ekranı	69
Şekil 4.9. MARIS, web yönetici arabirimi.....	70
Şekil 4.10. MARIS, web kullanıcı arabirimi giriş ekranı	71
Şekil 4.11. MARIS, web kullanıcı arabirimi, kullanıcı bilgileri	72
Şekil 4.12. MARIS Protege RDF veri modeli.....	75
Şekil 4.13. MARIS, RDF veri modeli mysql tabloları	76
Şekil 4.14. MARIS, RDFS veri modeli mysql tabloları	77
Şekil 4.15. MARIS, kullanıcı profili veri yapıları mysql tablo yapısı.....	78

RESİMLERİN LİSTESİ

Resim	Sayfa
Resim 4.1. Ericsson marka P990i model mobil cihaz	52
Resim 4.2. RoyalTek marka RBT 2210 model konum belirleme cihazı	53
Resim 4.3. MARIS mobil istemci uygulamasının başlatılması.....	62
Resim 4.4. MARIS mobil istemci uygulaması karşılama ekranı	63
Resim 4.5. MARIS mobil istemci uygulaması durum bilgisi ekranı.....	64
Resim 4.6. MARIS mobil istemci uygulaması menü ekranı.....	64
Resim 4.7. MARIS mobil istemci uygulaması ayarlar ekranı.....	65
Resim 4.8. MARIS mobil istemci uygulaması konum belirleme cihaz araması.....	65
Resim 4.9. MARIS mobil istemci uygulaması bluetooth cihaz arama sonucu	66
Resim 4.10. MARIS mobil istemci uygulaması durum bilgisi kayıt ekranı	67
Resim 4.11 MARIS mobil istemci uygulaması İnternet erişim izin ekranı	67
Resim 4.12. MARIS mobil istemci uygulaması servis bilgi ekranı	68
Resim 4.13. MARIS mobil istemci uygulaması yardım ekranı	68

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklama
ASCII	American Standard Code for Information Interchange
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communication
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
ISBN	International Standard Book Number
ITU	International Telecommunication Union
J2ME	Java 2 Micro Edition
JSP	Java Server Pages
OWL	Web Ontology Language
PC	Personal Computer
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
RMI	Remote Method Invocation
SOAP	Simple Object Access Protocol
UDDI	Universal Description Discovery and Integration

Kısaltmalar**Açıklama****URI**

Uniform Resource Identifier

URL

Uniform Resource Locator

URN

Uniform Resource Name

XML

Extensible Markup Language

W3C

The World Wide Web Consortium

WML

Wireless Markup Language

WSDL

Web Service Definition Language

WWW

World Wide Web

1. GİRİŞ

İnternet üzerinde çalışan bir servis olan Dünya Çapında Ağ (World Wide Web, kısaca www veya web) [1] ortaya çıktığı 1991 yılından beri yaygın olarak kullanılmakta ve hızla büyümektedir. Bu hızlı büyümenin temelinde; bilgi paylaşım isteği, web sayfalarının kolay oluşturulup yayınlanabilmesi ve oluşturulan sayfalara hızlı ve ucuz yollarla erişilebilmesi yatmaktadır.

Web'in ilk basamağı olarak kabul edebileceğimiz Statik Web, Tim Berners-Lee [2] tarafından 1989 yılında düşünülüp geliştirilmiştir. İstemci ve sunucu mimarisine göre yapılandırılan statik web, istemci olarak web tarayıcısı, sunucu olarak da genelde uzak bir yerde bulunan web sunucusundan oluşmaktadır.

Web sayfalarının İnternet üzerinde kolayca oluşturulup yayınlanabilmesi sonucu oluşan hızlı büyüme beraberinde bir çok problemi de getirmiştir. Sayfaların ilk kez oluşturulup yayımlandıktan sonra içeriklerinin zaman içerisinde bir çok defa değiştirilmesi istendiğinde yaşanan sıkıntılar ve bunların sonucunda oluşan yönetim zorlukları statik web yapısının daha da geliştirilmesi sonucunu doğurmuştur. Bu gereksinimler doğrultusunda dinamik web sayfası hazırlanabilmesi amacı ile bir çok çalışma gerçekleştirilmiştir. Çalışmaların çokluğu ve farklılığı nedeni ile zamana bağlı çalışma tanıtımı yapmak zordur. Ancak dinamik web sayfası hazırlanması düşüncesi Gary Kremen'e aittir. Kremen'in önerisi Amerika Patent Ofisi tarafından 13 Nisan 1999'da #5706434 numara ile patentleşmiştir [3].

2000'li yıllara doğru gelindikçe Web'in insanlar ve kurumlar için önemi de büyük bir hızla artmış, fiziksel olarak yapılan bir çok iş Web aracılığı ile elektronik ortamlara taşınmıştır (e-ticaret vb.). Bu gelişmeler, farklı kurumlar bünyesinde yer alan elektronik servislerin birbirleri ile haberleşerek bilgi işlemesi ve paylaşması gibi bir çok ihtiyacı da beraberinde getirmiştir [4]. Şubat 1998 tarihinde, platform bağımsız veri tanımlama standardı olan XML'in [5] W3C [6] tarafından resmi olarak duyurulmasıyla birlikte ihtiyaçları karşılayacak yeni teknolojilerin oluşmasının yolu da açılmıştır. Bir sonraki basamak olan XML yapısında oluşturulmuş mesajların

servisler arasında iletilmesini sağlayan protokol, Microsoft ve IBM tarafından geliştirilerek 2000 yılı yazında SOAP ismi ile duyurulmuştur. Arkasından 2000 yılının sonlarına doğru Web servislerini tanımlama dili olan WSDL ve Web servislerinin aranıp bulunması maksadı ile yayınlanmasını sağlayan UDDI belirtileri yine Microsoft ve IBM tarafından geliştirilerek piyasaya sunulmuştur. Kısa zamanda, Oracle, HP, Sun ve BEA gibi bilişim teknolojilerine yazılım altyapısı sağlayan diğer önemli firmalar da, web servis yapılarına desteklerini açıklayarak geliştirdikleri ürünlere gerekli standartları eklemişlerdir [7]. Böylece web servislerinin temelini oluşturan SOAP, WSDL ve UDDI standartları [8] web ortamında program geliştiricilerin hizmetine sunulmuş ve böylece web servisleri yapısı doğmuştur. Gelişme sürecine bakıldığında web servislerinin var olan standart Web'i destekleyici ve kullanılabilirliğini artırıcı özellikte olduğu görülmektedir. Bu nedenledir ki var olan web protokolleri üzerine kurulmuş ve açık XML standartlarına dayanmaktadır [4]. Lokal uygulamaların web uygulamalarına çevrilmesine olanak sağlayan ve İnternet üzerinde kullanılabilen Web API'leri de diyebileceğimiz web servisleri sayesinde farklı dillerde yazılmış uygulamalar birbirleri ile haberleşebilmekte ve veri işleyebilmektedirler. Bu özellikleri ile birlikte 2000 yılından günümüze kadar bir çok farklı teknolojik alanda kullanılmış ve halen de geliştirilmelerine devam edilmektedir.

Web servis uygulamaları sayısının gün geçtikçe artması ile birlikte Web, her geçen gün, statik web sayfaları havuzu olmaktan çıkıp, dinamik içerikler sunan servisler havuzu olmaya doğru ilerlemiştir [9]. Bu durum, her bir servisin işlediği verinin gösterim ve saklanma şeklinin belli bir standarda kavuşturulması düşüncesini ortaya çıkarmıştır. Çünkü web servisleri, servisler arasındaki iletişim ve veri aktarımı için gerekli standartları sağlamasına rağmen her bir servisin kendi içinde veri işleme ve saklama şekli hakkında herhangi bir yöntem sunmamaktadır. Bu, Web üzerinde bulunan verilerin bilgisayarlar arasında paylaşıp yorumlanabilmesi için her seferinde yeni web servislerinin oluşturulmasına yol açmaktadır. Ayrıca standart web sayfaları düşünüldüğünde de sayfalar içindeki bilgiler sadece insanlara hitap edecek şekilde gösterilmektedir. Bahsedilen noktaları dolduracak ve mevcut Web'in bir uzantısı olarak Web üzerinde yer alan bilgilerin anlamsal olarak sınıflandırılabilmesi

için gerekli yöntem ve standartları içerecek Anlamsal Web yapısı 2001 yılında Tim Berners-Lee tarafından önerilmiştir [10]. Anlamsal Web, en yalın anlatım ile anlam kazandırılmış web olarak açıklanabilir. Bilgilerin meta verilerinin (verilerin verileri) oluşturulmasıyla bilgisayar ortamında bulunan uygulamaların, insan etkileşimi olmadan bu bilgileri yorumlayabilmesi ve ihtiyaca yönelik olarak kullanabilmesi hedeflenmektedir. Anlamsal Web'in işlevsellik kazanması ile birlikte alışılmış geleneksel bilgi paylaşım yaklaşımlarının değişeceğine inanılmaktadır. Sunucuların, kişisel bilgisayarların, cep bilgisayarlarının ve daha bir çok elektronik cihazın yorumlamalı olarak bilgi alışverişi yapıp hayatı kolaylaştıracağı düşünülmektedir.

İnternet teknolojilerinin gelişmesi ve bununla birlikte Web kullanımının yaygınlaşmasının diğer bir sonucu da bu alanda kullanılan cihazların çeşitlenmesidir. Zamandan ve yerden bağımsız olarak kullanılabilen mobil cihazlar bu çeşitlenmenin en güzel örneğidir. Mobil cihazlar, kullanılmaya başlandığı 1990'lı yılların başlarından itibaren gün geçtikçe daha fazla kişiye ulaşırken bir taraftan da kullanıcılara sundukları teknoloji olarak gelişmektedirler [11]. International Telecommunication Union (ITU) verilerine göre Ağustos 2007 itibarı ile dünya genelinde cep telefonu kullanıcı sayısı 3 milyarın üzerine çıkmıştır [12]. Telekom ve mobil medya pazarında tüketiciler üzerinde araştırma çalışmaları yapan Nielsen Mobile firması [13] rakamlarına göre Amerika'daki 350 milyar dolarlık mobil sektörü ve bu sektör içerisinde 237 milyon kullanıcı bulunmaktadır. 2007 yılının Eylül ayı içerisinde bu kişilerin %70'i telefonlarını kısa mesaj göndermek, 32 milyonu İnternet'e erişmek ve %41'i de resimli mesaj göndermek için kullanmışlardır [14]. Mobil cihaz teknolojisi geliştirilmeye başlandığı zaman itibarı ile aradan geçen kısa sürede çok yol almıştır. İlk zamanlarda bulunduğu sadece ses iletişimi sağlama noktasından gelişerek ilerlemiş ve özellikle İnternet gibi istenildiği an erişilip kullanılabilen yeni özellikleri bünyesine katmıştır [15].

Mobil teknolojilerdeki hızlı gelişmeler sonucunda geleneksel bilgisayar uygulamaları yerini gün geçtikçe mobil uygulamalara bırakmaktadır. Mobil uygulamalar her zaman ve her yerde erişilip kullanılabilirler. Bu hareketliliğin sonucunda oluşan çevresel bilgiler ile kullanıcıya ait kişiselleştirilmiş bilgilerin beraber

işlenilmesi sayesinde kullanıcılara daha faydalı uygulamalar sunulabilmektedir. Bu kapsamda işlenen kullanıcı profili, bulunulan yer, zaman ve çevresel veriler gibi kişi veya diğer ilgili varlıklara özgü dinamik bilgilere genel anlamda filtrelenmiş içerik yada bağlam denir. Filtrelenmiş içeriği, uygun veri yapıları formatında saklanan diğer statik bilgiler ile birlikte akıllı yöntemlerle işleyip kullanıcı için bilinçli sonuçlar üreten uygulamalara da bağlam bilinçli uygulamalar adı verilir.

Bağlam bilinçli uygulamalar kavramı ilk olarak 1994 yılında Schilit ve Theimer [16] tarafından ortaya atılmıştır. Onlar, bağlam bilinçli uygulamaları; kullanıldıkları yer, yakınlarında bulunan diğer kişiler veya varlıklar ve ayrıca bu varlıkların zamana bağlı değişimlerine uygun olarak kendisini uyarlayabilen yazılımlar olarak tanımlamışlardır. Bundan sonra da bir çok defa bir çok kişi tarafından içerik bilinçli uygulamalar kavramı tanımlaması yapılmıştır. Bunların çoğu spesifik anlam içeren tanımlamalardır [17].

Bağlam bilinçli uygulamalar çoğunlukla mobil cihazlar ve mobil kullanıcılar için düşünülmelidir. Çünkü filtrelenmiş içerik olarak tanımlanan parametreler daha çok mobil platformlarda oluşabilmekte ve uygulamalara girdi olarak sağlanabilmektedir. Dolayısı ile mobil ortamlar için geliştirilen bağlam bilinçli servis uygulamaları; yer, zaman, çevredeki kişiler, çevredeki varlıklar ve kişilere özgü aktiviteler gibi filtrelenmiş içerikleri akıllı yöntemlerle işleyerek kullanıcılara bilinçli ve dinamik sonuçlar sunmaktadırlar [18].

Filtrelenmiş içerik elde edilmesinde en çok yararlanılan bilgi yer bilgisidir. Yer bilgisi kullanılarak hazırlanan uygulamalara genel olarak yer bazlı servisler denilmektedir [19]. Yer bilgisi daha çok gezinim ve yön bulma amaçlı kullanılmaya başlanmış olmasına rağmen günümüzde gelişmiş mobil uygulamalarda da yerini almaktadır. Özellikle coğrafi bilgi sistemleri, telematik, turistik amaçlı sistemler ve lojistik bilgi sistemleri alanlarında geliştirilen uygulama sayıları her geçen gün artmaktadır [20]. Yer bilgisinin oluşturulmasında kullanılan teknolojilere yer belirleme teknolojileri denir. Bu teknolojiler belirleme yöntemine göre iki ana grupta değerlendirilebilir.

1. *Küresel Yer Belirleme Sistemi*: 1978 yılında Amerika Savunma Bakanlığı tarafından askeri amaçlı yön belirleme, gezinim ve ateşli silahların hedef saptaması gibi alanlarda kullanılmak üzere geliştirilmesine başlanmıştır. Nisan 1995 yılında 24 uydudan oluşan sistemin geliştirilmesi tamamlanmıştır. Sistem, yeryüzüne düşük güçlü radyo sinyalleri aracılığı ile düzenli olarak kodlanmış bilgi gönderen uydu ağından oluşmaktadır. Temel çalışma prensibi; yeryüzünde bulunan alıcıların kodlanmış bilgi içeren sinyalin ulaşma zamanından sinyali aldıkları uydunun uzaklığını hesaplamalarına dayanır. 1984 ve 2000 yıllarında Amerika başkanlarının verdikleri kararlar doğrultusunda Küresel Yer Belirleme Sistemi sivil kullanıma da açılmıştır. Böylece ticari anlamda geliştirilen projelerde sistemin kullanımı hızlı bir şekilde yaygınlaşmaya başlamıştır [19]. Acil durumlarda kullanılması amacı ile Amerika'da Küresel Yer Belirleme özelliği telefonlarda bir standart haline getirilmiştir ve 2011 yılına kadar bu özelliğe sahip 720 milyon cihaz satılacağı düşünülmektedir [21]. Türkiye'de de özellikle Nokia ve Sony Ericsson firmalarının piyasaya sundukları Küresel Yer Belirleme özelliğine sahip mobil telefon seçenekleri ile son zamanlarda kullanım yaygınlaşmaktadır.
2. *GSM Hücre Bilgisi ile Yer Belirleme Sistemi*: GSM mobil telefon ağı doğası gereği hücresel yapıdadır. Bu hücresel yapı sayesinde değişik teknikler kullanılarak GSM mobil telefonlarının yerlerinin belirlenmesi mümkündür. Temel çalışma prensibi; mobil telefonun kayıt olduğu baz istasyonunun belirlenmesine dayanır [19].

Günümüzde mobil platformlardan elde edilen dinamik verilerin (yer, zaman vb.) diğer standart veriler ile birlikte web teknolojilerinde ulaşılan son nokta olan anlamsal web teknikleri kullanılarak işlenilmesi yöntemine dayanan bir çok çalışma yapılmaktadır. Bunun sonucunda mobil kullanıcıların sahip oldukları esneklik ve dinamikliğe uygun servisler geliştirilebilmekte ve kullanıma sunulmaktadır.

Olivetti araştırma laboratuvarlarında geliştirilen ve aktif kimlik sistemi adı verilen proje, bağlam bilinçli uygulamalara verilebilecek ilk örneklerden biridir [22]. Özel algılayıcı sistemi sayesinde kişilerin yer bilgisi merkezi sisteme gönderilmekte bu

yer bilgisine göre telefon çağrıları yönlendirilmektedir [23]. 1994 yılında Xerox Palo Alto araştırma merkezinde geliştirilen ve ParcTab sistem olarak adlandırılan proje ilk olarak bağlam bilinçli uygulamalar kavramının ortaya atıldığı diğer bir örnektir [22]. ParcTab sistemi PDA cihazlarında çalışmakta ve yazıcı gibi bir çok ofis kaynağının yakınlığını belirtmektedir [24].

MIT (Massachusetts Institute of Technology) tarafından geliştirilen Oxygen projesi, CMU (Carnegie Mellon University) tarafından geliştirilen Aura projesi ve University of California Berkely tarafından kurulan GUIR araştırma grubu içerisinde geliştirilen çeşitli farklı projeler bağlam bilinçli uygulamalara verilebilecek başlıca güncel örneklerdir [25].

Mobil platformlar için Anlamsal Web teknikleri kullanılarak geliştirilen bağlam bilinçli uygulama alanlarından en önemlisi rehberlik ve bilgilendirme servislerine dayanan sistemlerdir. Bu tür servisler genel olarak yer bilgisini işlemeye yöneliktir. Norman Sadeh ve arkadaşları tarafından hazırlanan müze tur rehberi uygulaması, müze ziyaretçilerinin tur zamanlarını en etkin şekilde kullanmalarını hedeflemektedir. Ziyaretçilere müze girişinde cep bilgisayarları dağıtılmakta ve ilk olarak yaş, ilgi alanları ve tur zamanları gibi bazı kısa sorulara cevap vermeleri istenerek kullanıcı profilleri oluşturulmaktadır. Böylece ziyaretçilerin ilgi alanları tespit edilmekte ve tur boyunca yapılacak yönlendirme ve rehberlik hizmetlerinde bu bilgiler kullanılmaktadır. İlgi alanlarına göre müzenin ilgili bölümlerine yönlendirmeler yapılmakta, tur zamanından arda kalan süreler için profildeki sevilen yemekler için civardaki tanınan uygun lokantalar önerilebilmektedir. Ayrıca bu hizmetler karşılığında ziyaretçilerden alınan görüşler ve uygun geri bildirimler daha sonraki ziyaretçilere aktararak deneyimlerin paylaşılması sağlanmaktadır [25].

Diğer bir örnek olan SMOP projesi Ege Üniversitesi Bilgisayar Mühendisliği bölümünden Özgür Gümüş ve arkadaşları tarafından geliştirilmiştir. SMOP, yer bilgisi duyarlı, Anlamsal Web tabanlı mobil bir uygulamadır. Uygulama genel olarak üç katmandan oluşmaktadır. Mobil istemci; yer bilgisini oluşturmak, kullanıcının isteklerini girdi olarak yazabileceği arabirimi sunmak ve bu bilgileri sunucuya

gönderdikten sonra sunucudan gelen sonuçların kullanıcıya gösterildiği arabirimi oluşturmak işlemlerinden sorumludur. Sunucu; istemciden gelen istekleri karşılamak, gerekli cevapları platform web servisleri aracılığı ile değerlendirip oluşturmak ve bunları istemciye geri göndermekten sorumludur. Platform servisleri; sunucu tarafından iletilen istekleri anlamsal web kapsamında değerlendirip işleyerek uygun cevapları oluşturmaktan sorumludur [26].

Carnegie Mellon Üniversitesi Mobile Commerce Lab tarafından geliştirilen MyCampus projesi Anlamsal Web teknikleri kullanılarak geliştirilen bağlam bilinçli uygulamalara verilebilecek diğer güncel bir örnektir. MyCampus, kampüs yaşantısını kolaylaştırmayı amaçlayan ve içerisinde bir çok servisi barındıran bir Anlamsal Web platformudur. Kullanıcılar platform içerisinde bulunan; toplantı düzenleme, doküman paylaşma, aktivite organizasyonu ve e-posta mesaj yönetimi gibi çeşitli servislere abone olabilmekte ve bu servisler üzerinden elektronik asistanlık hizmete alabilmektedirler [27].

Mobil platformlar için Anlamsal Web teknikleri kullanılarak geliştirilen uygulama alanlarından bir diğeri de arama sistemleridir. Joo-Seong Jeon ve Gi-Jeong Lee tarafından geliştirilen Anlamsal Web tabanlı mobil arama sistemi geleneksel arama yöntemleriyle karşılaştırıldığında arama işleminin %20 oranında iyileştiği görülmüştür. Ayrıca geliştirilen sistem sayesinde dinamik arama, akıllı cümle araması ve benzerlik araması gibi yeni nesil arama özelliklerinin de başarılı bir şekilde kullanılabileceği gösterilmiştir [28].

Bu tez çalışmasında ise geliştirilen mobil uygulama aracılığı ile kullanıcıya ait yer bilgisi gerçek zamanlı olarak elde edilerek merkezi sisteme gönderilmektedir. Merkezi sistemde kullanıcı profil bilgileri ve anlamsal web yöntemleri ile modellenmiş kullanıcıya sunulacak servislere ait veriler bulunmaktadır. Bu veriler, merkezi sisteme iletilen yer bilgisi ile beraber değerlendirilerek ilgili servis için içerik bilinçli sonuçlar üretilmekte ve kullanıcıya sunulmak üzere mobil uygulamaya gönderilmektedir.

2. ANLAMSAL WEB

Web alanında yıllar boyunca meydana gelen gelişmeler ve değişimler, günümüz web teknolojilerinde en son ulaşılmış nokta olan Anlamsal Web'in doğuşunda rol oynamış önemli tetikleyici unsurlardır. Bu temel unsurlar aşağıda yer alan Çizelge 2.1'de gösterilmiştir.

Çizelge 2.1. Web teknolojileri evrimi [29]

	Statik	Dinamik	Söz Dizimi	Anlamsal
Kodlama	HTML	+ RDBMS	+ XML	+ RDF/OWL
Oluşturma	El ile	Sunucu tarafı uygulamalar aracılığıyla	Şema tabanlı uygulamalar aracılığıyla	Model tabanlı uygulamalar aracılığıyla
Kullanıcı	İnsan	İnsan	İnsan ve Uygulama	İnsan ve Uygulama
Kullanım	Tarayıcı	Oluşturma, Sorgulama, Güncelleme	Bütünleştirme	Birlikte Çalışma
Uygulama	Tarayıcılar	Tarayıcılar	Bütünleşik Prosesler, Kurumsal Bütünleşik Uygulamalar, İş süreçleri yönetimi yazılımları	Zeki Ajan yazılımları, Anlamsal çekirdek yazılımlar

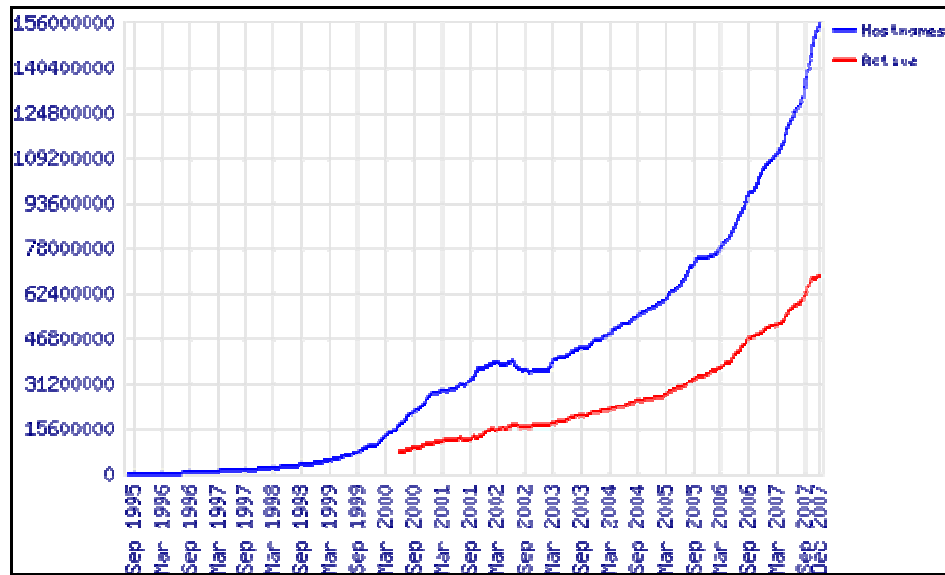
Web üzerinde yayınlanan bilgilerin, tarayıcılar aracılığı ile insanlar tarafından erişilebilmenin yanı sıra sistemler ve uygulamalar arasında entegre bir şekilde işlenebilmesi amacıyla bilgisayarlar tarafından da kullanılacak şekilde tanımlanması mümkündür. Bilgilerin bilgisayarlar tarafından kullanılması ve otomatik olarak işlenmesinin bir yolu bilgilerin bilgisayarların da anlayabileceği şekilde oluşturularak modellenmesidir. İşte bu düşünce tam olarak Anlamsal Web'in amacını ortaya koymaktadır. Yani Web üzerinde bulunan bilgilerin bilgisayarlar tarafından işlenebilmesinin mümkün kılınması [29].

Bu kapsamda Berners-Lee (2001) şöyle demektedir [29].

“Anlamsal Web, farklı bir Web çeşidi olmayıp, mevcut Web'in bir uzantısıdır. Bilgisayar ve insanların birlikte çalışmalarını kolaylaştırmak için bilgilerin ve anlamlarının uygun şekilde tanımlanmasıdır”.

2.1. Web

Web, kullanılmaya başlandığı 1991 yılından beri hızlı bir şekilde büyümüş ve dünya genelinde yaygınlaşmıştır. Bu artış özellikle son yıllarda çok daha büyük bir ivme ile gerçekleşmektedir. Web sayfaları üzerinde istatistiksel bilgiler oluşturan netcraft.com'un 2007 yılı sonunda yayınladığı Ağustos-1995 ve Aralık-2007 tarihleri arasını kapsayan istatistiklerine göre günümüzde 155 milyona yakın web sayfası bulunmaktadır. Bunlardan 40 milyonu ise son bir yılda hizmete girmiştir [30].



Şekil 2.1. Web sayfalarının artış grafiği [30]

Web'in kullanım artışına paralel olarak bir taraftan da teknolojik gelişimi sürdürülmektedir. Günlük hayatta yapılan işler Web üzerine taşınmak istendikçe web teknolojilerinden beklentiler de bu oranda artmaktadır. Beklentilerin karşılanması için başlangıçtan itibaren sürekli çalışmalar yapılmıştır ve yapılmaya da devam edilmektedir. Bu kapsamda 1990'lı yılların başlangıcında geliştirilen Web'e statik web denilmektedir. 1990'lı yılların sonlarına doğru geliştirilen ve kullanımdaki mevcut Web'e dinamiklik kazandıran teknolojiye ise dinamik web adı verilmektedir. Web'in temellerinin atıldığı bu teknolojiler genel olarak Web 1.0 olarak da adlandırılmaktadır.

2.1.1. Statik web

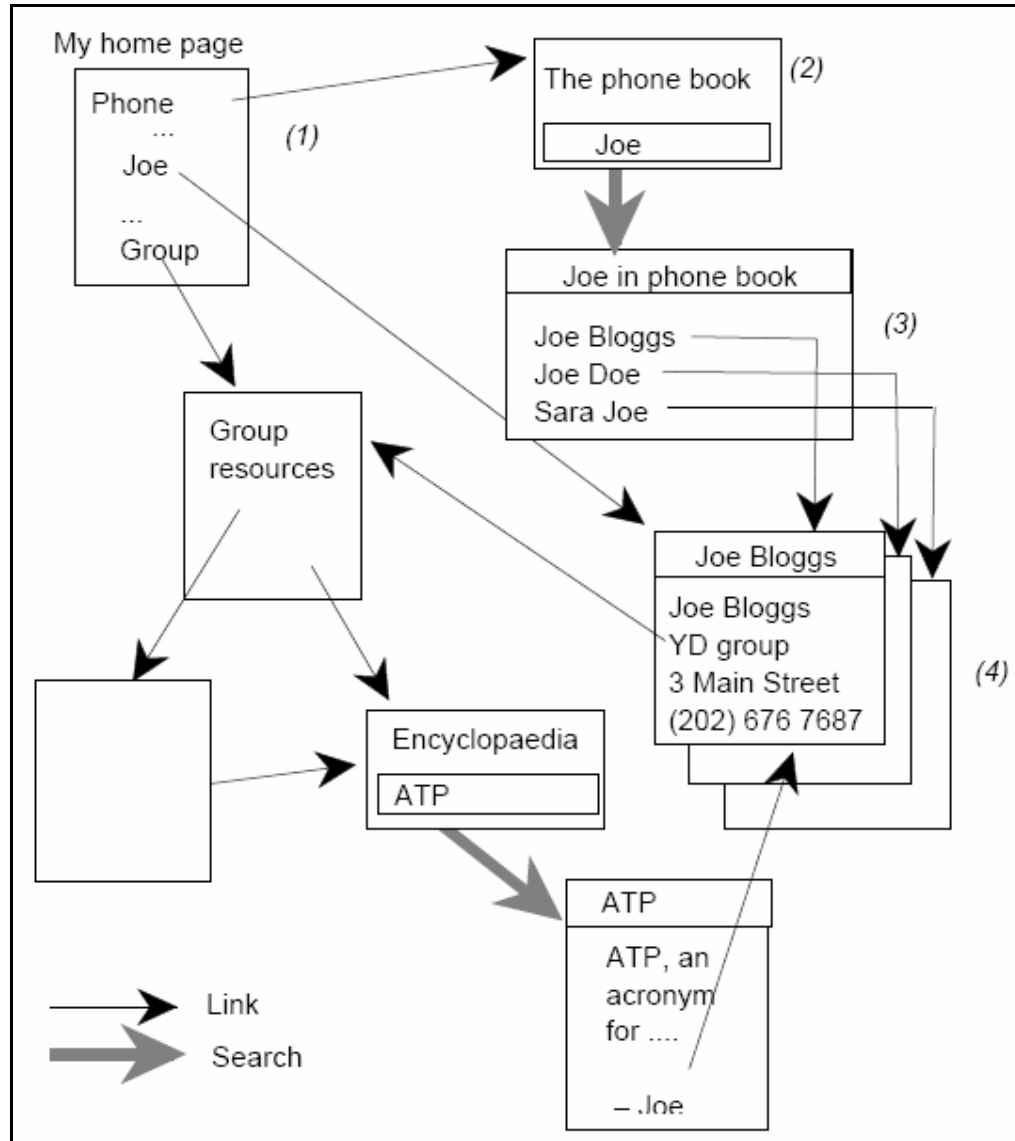
Statik Web, Tim Berners-Lee tarafından 1989 yılında düşünülüp geliştirildikten sonra hızla büyüüp dünya çapında bir çok şirketin, üniversitenin, organizasyonun ve son kullanıcıların vazgeçilmez bilgi aktarım ve paylaşım aracı olmuştur.

Statik web, mimari olarak istemci ve sunucu yapısına dayanmaktadır. İstemci olarak web sayfalarının görüntüleneceği platformlarda çalışan ve web tarayıcısı adı verilen yazılımlar kullanılmaktadır. Sunucular ise genelde istemcilerden uzak bir yerde bulunan ve web sunucusu adı verilen yazılımlardan oluşmaktadır. Web sunucusu, basit bir işaretleme dili olan HTML aracılığı ile oluşturulan ve web sayfalarının içeriğini tutan dosyaları barındırmaktadır [31]. Sunucularda barındırılan bu kodlanmış dosyaların kullanıcı tarafından okunabilir olarak gösterilmesi sürecinde gerçekleşen aşamalar aşağıdaki gibidir.

1. Web tarayıcısı kullanıcıdan aldığı sayfa görüntülenme isteğini inceler ve hangi sunucudan hangi dosyayı istemesi gerektiğini çıkarır.
2. İstemci, ilgili sayfa için ilgili sunucudan bir haberleşme protokolü olan HTTP aracılığı ile istekte bulunur.
3. Gelen isteği alan web sunucusu isteği inceler ve istekte bulunulan sayfayı kodlanmış olarak istemciye göndermek üzere hazırlar. Kodlanmış dosya yine HTTP aracılığı ile istemciye iletilir.
4. Dosyayı alan web tarayıcısı HTML dilinde oluşturulmuş kodları çözerek dosya içerisindeki bilgileri kullanıcıya gösterir.

Genel olarak her bir kaynak için farklı HTML dosyası oluşturulmakta ve bir kaynaktan diğer bir kaynağa geçmek için hipermetin(üstmetin) bağlantıları kurulmaktadır. İlgili Web sayfasını görüntülemek isteyen kullanıcı ilk olarak ana sayfa adı verilen başlangıç sayfasına erişmektedir. Bu sayfada bilgiler içeren düz metinlerin yanı sıra kullanıcının görüntülemek isteyebileceği diğer sayfalar için de ilgili bağlantılar yer almaktadır. Kullanıcı bu bağlantıları kullanarak diğer sayfalarda

gezinti yapabilmektedir. Bağlantıların yanı sıra ayrıca arama fonksiyonu kullanılarak da ulaşmak istenilen sayfaya gidilebilmektedir [1].



Şekil 2.2. Web veri modeli [1]

2.1.2. Dinamik web

Statik web kullanımının hızlı bir ivme ile artması sonucu web teknolojilerinin daha fazla geliştirilmesi ihtiyacı da kısa sürede ortaya çıkmıştır. Çünkü bu hızlı büyüme sonucu son kullanıcılar ve geliştiriciler bir çok problem ile karşı karşıya kalmıştır. Web sayfalarının oluşturulma şekillerinden dolayı yayınlandıktan sonra değiştirilebilmesi için insan müdahalesine ihtiyaç duyulması bu problemlerin başında gelmekteydi. Bu tür kısıtlamalar bir yandan web sayfalarının yönetimini zorlaştırırken bir yanda da içeriğin zenginleşmesini ve dinamikliğini engellemekteydi [3]. Spor sayfaları, borsa sayfaları ve sanal mağaza sayfaları gibi bir çok web sayfası içeriğinin dinamik oluşturulup yayınlanması istenmekteydi [32].

Gary Kremen'in dinamik web teknolojisi icadı ile birlikte tarayıcıların yorumladığı HTML dili ile oluşturulmuş dosyaların statik olarak saklanıp her seferinde aynı içeriğin istemcilere gönderilmesi yerine web sunucudan bağımsız olarak içeriğin oluşturulması ve bu içerik doğrultusunda HTML dil dosyasının yeniden yapılandırılarak istemcilere gönderilmesi sistemi doğmuştur. Bu yapı ile birlikte uygulama kodları ile içerik verileri de birbirinden ayrılmaya başlanmıştır (Web sayfalarına içerik oluştururken veritabanlarının kullanılması). Günümüzde bu amaçlar doğrultusunda kullanılan popüler yorumlamalı programlama dilleri; PHP, PERL, ASP ve JSP, ASP.NET'dir.

Dinamik web sistemine göre çalışan bir web sunucu ile web tarayıcı arasında işleyen süreç aşağıdaki gibidir [3].

1. Web tarayıcısı kullanıcıdan aldığı sayfa görüntülenme isteğini inceler ve hangi sunucudan istekte bulunacağını belirler.
2. İstemci ilgili web sunucudan istekte bulunur.
3. Web sunucu isteği inceler ve isteğin türünü belirler. Statik HTML dosyası mı isteniyor yoksa dosya dinamik olarak bir uygulama aracılığı ile mi oluşturulacak.
4. Eğer istek statik html dosyası ise, web sunucu dosyayı hazırlar ve istemciye geri dönüş olarak gönderir.

5. Eğer istek dinamik oluşturulması gereken bir içerik ise, web sunucu içeriği oluşturacak uygulamayı belirler.
6. Uygulama çalışır ve sonuç olarak HTML ile kodlanmış dosyayı üretir.
7. Dinamik içeriği barındıran dosya web sunucu aracılığı ile geri dönüş olarak istemciye gönderilir.

2.2. Web Servisleri

Web 2.0 olarak da adlandırılan web servis teknolojileri 2000’li yılların başlarında kullanılmaya başlanmış ve başta Microsoft ve IBM olmak üzere bir çok yazılım firması tarafından desteklenerek kısa sürede web ortamında program geliştiricilerinin yoğun olarak kullandıkları bir platform haline gelmiştir.

Web servisleri, açık XML standartlarına dayanan ve hali hazırda mevcut olan Web protokolleri üzerine oturtturularak uygulamalar arası haberleşmeyi sağlayan sistematik ve genişleyebilir yapılardır [4]. Bu yapılar aracılığı ile oluşturulan uygulamalar aşağıdaki özellikleri sağlarlar [7].

1. Uygulama, içerdiği servislere ait fonksiyonları ve bu fonksiyonların kullanım detaylarını açık olarak tanımlar. Böylece ilgili servisleri kullanmak isteyen diğer uygulamalar fonksiyon tanımlarını ve kullanım detaylarını anlayabilirler.
2. Uygulama, sunduğu servislerin diğer uygulamalar tarafından istenildiğinde aranıp bulunabilmesine izin verir. Böylece gerektiğinde ilgili servisler kullanılabilir durumda olur.
3. Uygulama, içerdiği servislere ait fonksiyonların, diğer uygulamalar tarafından istenildiğinde kolayca çağrılabilmesini sağlar.

Web servislerinin bu özellikleri yerine getirirken yararlandıkları temel teknolojiler aşağıdadır [7].

1. *Platform Bağımsızlık*: Bütün yazılım tedarikçilerinin altyapıları aynı belirtilmeleri desteklemelidir.
2. *Dilden Bağımsızlık*: Herhangi bir yazılım dilinde geliştirilen bir uygulama web servisine dönüştürülürken gerçekleştireceği belirtiler bakımından diğer bütün yazılım dilleri ile aynı yöntemi kullanmalıdır.
3. *Kısıtsız Yayılım*: Web servis teknolojilerini geliştiren yazılımcıların en önemli amacı en geniş şekilde kısıtsız olarak yayılım olmalıdır.

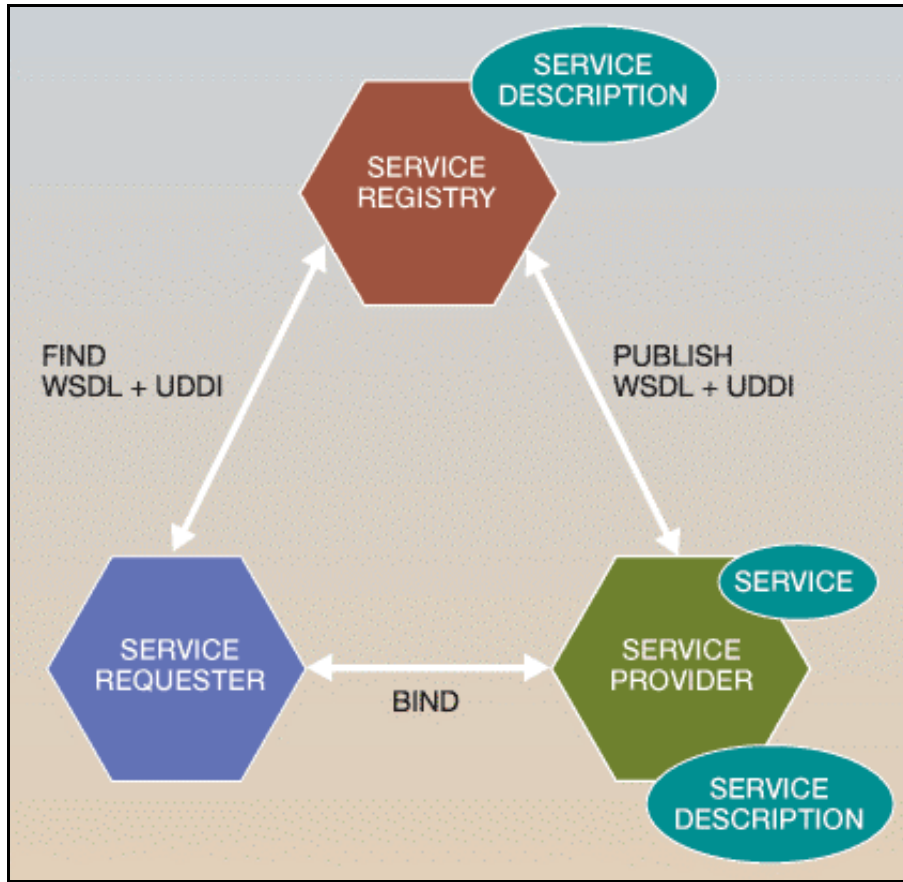
2.2.1. Web servis teknolojileri

Yukarıda da belirtildiği gibi bir web servisi kendisini tanımlayabilen, diğer uygulamalar tarafından arandığında bulunabilen ve kolayca çağrılabilen yazılım yapısıdır. Web servisin kendisini tanımlayabilmesi WSDL teknolojisi, arandığında bulunabilmesi UDDI teknolojisi ve kolayca çağrılabilmesi de SOAP teknolojisi ile mümkün olmaktadır. Her üç teknoloji de temel olarak geleneksel veri tanımlama standardı olan XML üzerine kuruludur [7].

WSDL, UDDI ve SOAP teknolojileri Şekil 2.3’de gösterilen üç ana birimin birbirleri ile etkileşimi sırasında kullanılırlar. Bu üç ana birim ve görevleri aşağıdaki gibidir [33].

1. *Servis Sağlayıcı (Service Provider)*: Servis sağlayıcı istemcilerin sağlayıcıda bulunan servislere erişimini sağlar. Servis sağlayıcı kendisinde bulunan web servislerinin tanımını servis kayıt birimine kaydederek ilgili servislerin nasıl çağrılacağı belirtir. Bu kaydetme işleminde WSDL ve UDDI teknolojileri kullanılır.
2. *Servis İstemci (Service Requestor)*: Servis sağlayıcısında bulunan web servislerini çağırarak kullanan istemci uygulamalardır. Web servisinin nasıl çağrılacağını ve ilgili parametreleri servis kayıt biriminden arayarak bulur ve ister. Servis istemci ile servis sağlayıcı arasında gerçekleşen web servislerinin işletilmesi sürecinde SOAP teknolojisi kullanılır.

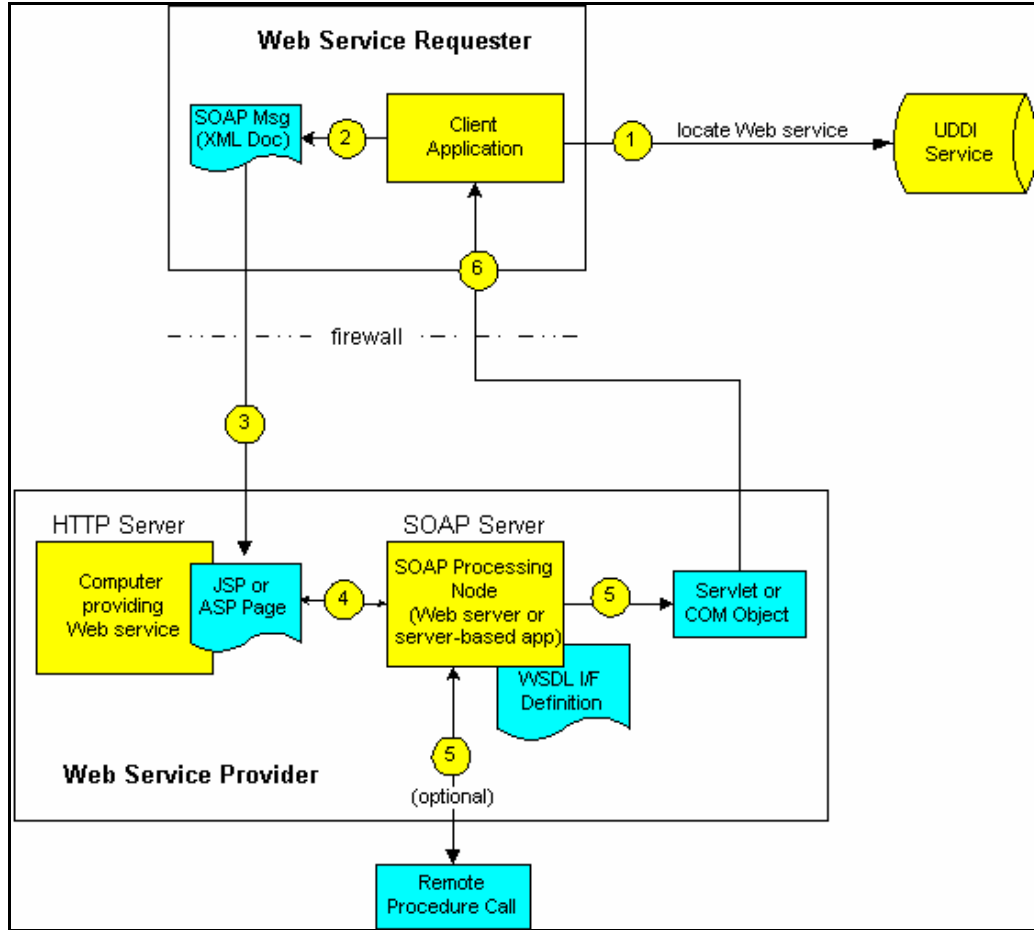
3. *Servis Kayıt Birimi (Service Registry)*: Servis sağlayıcılarının yayınladıkları web servisi tanımlarını saklar ve aranıp bulunmasını sağlar. Servis istemcileri servis kayıt birimini tarayarak istediği servisler hakkında bilgi alabilir. Servis kayıt birimi her servisin nasıl çağırılacağı konusunda tanım bilgilerini içerir. Servis istemcilerinin servis kayıt biriminden ilgilendikleri web servisleri hakkında bilgi edinme sürecinde UDDI teknolojisi kullanılır.



Şekil 2.3. Web servis temel modeli [34]

Servis istemci tarafından web servislerinin aranıp bulunması ve daha sonra çağrılarak kullanılması aşamasında gerçekleşen süreçler Şekil 2.4'de gösterilmiş ve aşağıda açıklanmıştır [33].

1. Servis istemci, servis kayıt biriminden ilgilendiği web servisini arar ve bulur. Böylece bu web servisini nasıl çağıracağını ve nasıl kullanabileceğini daha sonraki adımlarda yararlanmak üzere öğrenmiş olur.
2. Servis istemci birinci adımda öğrendiği parametreler doğrultusunda XML standartlarına göre bir SOAP mesajı hazırlar.
3. İstemci SOAP mesajını web sunucusu veya uygulama sunucusunda çalışan SOAP istek dinleyicisine gönderir. İstek dinleyici gelen isteklere cevap veren sunucu programlardır. Bu programlar bir JSP, ASP, CGI veya ISAPI uygulaması olabilir.
4. İstek dinleyici gelen isteği SOAP sunucusuna yönlendirir. SOAP sunucusu gelen SOAP mesajını çözümler ve gerekli parametreleri göndererek istenen nesnenin istenen yöntemini çağırır.
5. Çağırılan nesnedeki yöntem çalışır ve sonuçları SOAP sunucusuna gönderir. SOAP sunucusu gelen sonucu SOAP mesajı formatında biçimlendirerek istemciye gönderir.
6. İstemci gelen SOAP mesajının içindeki bilgileri alarak istekte bulunan programa gönderir.



Şekil 2.4. Servis istemci ve servis sağlayıcı süreçleri [35]

Extensible Mark-up Language (XML)

XML, bilgisayar ortamındaki mesajların tanımlanmasında kullanılan belirtimleri sunan bir standarttır. Geliştirilen uygulamalar bu standart sayesinde birbirleri arasında kolayca mesaj iletimi yapabilmektedirler. Bunun için tag (etiket) olarak ifade edilen işaretler kullanılmaktadır [7]. Mesaj içerisindeki veriler bu etiketler sayesinde işaretlenmektedir. Böylece mesaj standart bir formatta sunulabilmektedir.

```

<musteri>
  <isim>Osman Kunduraci</isim>
  <adres ulke="TR">
    <cadde>38. Cadde</cadde>
    <mahalle>zeytinlik</mahalle>
    <sehir>Artvin</sehir>
    <postakodu>08000</postakodu>
  </adres>
  <durum satinalma="2" son-satinalma="12-05-2006" />
</musteri>

```

Şekil 2.5. XML işaretleme örneği [7]

Şekil 2.5’de görülen, *<musteri>* etiketi ile açılıp *</musteri>* etiketi ile kapatılan mesaj bir müşteriye ait içeriği göstermektedir. Görüldüğü gibi veriler etiketler yardımı ile yapılandırılmıştır. Bu sayede mesaj, insanlar tarafından okunabilir olmanın yanında aynı zamanda bilgisayarlar için de okunabilir olmaktadır [7]. Bilgisayarlar için okunabilir olması demek, farklı uygulamalar arasında veri haberleşmesinin de yapılabilmesi demektir. XML, kolay kullanılışı ve uygulanışı sayesinde kısa sürede yazılımlar arasında veri haberleşme standardı olarak yerini almıştır. Bu sayede dağıtık uygulamaların gelişmesi için de her zaman temel taşlardan birisi olmuştur.

XML yapısı verinin gösterilmesinden ziyade daha çok verinin taşınması görevini yerine getirmektedir. Bununla birlikte son zamanlarda verinin saklanması ve tanımlanması işlemlerinde de gittikçe kullanımı yaygınlaşmaktadır. Özellikle yazılım alanlarında yaşanan teknolojik gelişmeler XML standartlarını bir temel eleman olarak kullanmaktadır.

Simple Object Access Protocol (SOAP)

XML ile birlikte yazılım bileşenlerinin birbirleri arasında gönderip aldıkları mesaj içeriklerinin standart bir şekilde tanımlanabilmesi sağlanmıştır. Temelde bu özellikten faydalanılarak geliştirilen SOAP sayesinde yazılım bileşenlerinin birbirleri arasında mesaj geçerek birbirlerinin ilgili fonksiyonlarını çağırabilmeleri mümkün olmuştur [7]. SOAP teknolojisinin farklı yazılım geliştirme dillerine ve

platformlarına adapte edilmesi ile birlikte de dilden bağımsız bir uygulamalar arası haberleşme standardı doğmuştur.

SOAP protokolü temel olarak aşağıdaki yapıları içermektedir [7].

1. Yazılımlar arası haberleşme sırasında kullanılan mesajın içeriğini ve nasıl işleneceğini tanımlayan bir arabirim.
2. Mesaj içerisinde kullanılan veri yapılarının tanımlamaları için oluşturulmuş kodlama kuralları.
3. Yazılımın hangi fonksiyonlarının diğer yazılımlar tarafından çağrılıp kullanılabileceğine ve çağırılma sonucu fonksiyonda oluşan cevabın geri dönüşüne yönelik kurallar.
4. Mesaj taşınırken kullanılacak haberleşme protokolünün nasıl kullanılacağına dair kurallar.

SOAP, bir istekte bulunma ve cevaplama mekanizması kullanmaktadır. Bir yazılım bileşeni diğer bir yazılım bileşenine istekte bulunur ve istek sonucu oluşan cevabı geri dönüş olarak alır. İstek ve cevap mesajları XML standartlarına göre oluşturulur ve taşınır [7].

Şekil 2.6’da SOAP istek mesajı örneği görülmektedir. Satır başlarında bulunan numaralar gerçek mesajın içeriğinde bulunmayıp, örnekte açıklamalar için referans olarak kullanılmıştır.

```

1. POST/StockQuote HTTP/1.1
2. Host:www.stockquoteserver.com
3. Content-Type: text/xml; charset="utf-8"
4. Content-Length: nnnn
5. SOAP Action: "Some-URI"

6. <SOAP-ENV:Envelope
7.   xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
8.   SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
9.   <SOAP-ENV:Body>
10.    <m:GetLastTradePrice xmlns:m="Some-URI">
11.      <symbol>STOCKSYMBOL</symbol>
12.    </m:GetLastTradePrice>
13.  </SOAP-ENV:Body>
14. </SOAP-ENV:Envelope>

```

Şekil 2.6. SOAP istek mesajı örneği [7]

İstekte bulunan yazılım bileşeni istekle birlikte gelen hisse senedi tanıtıcı numarasına göre ilgili hisse senedine ait fiyat bilgisini cevap olarak geri göndermektedir. Mesajda bulunan satırlara ilişkin açıklamalar aşağıdadır [7].

1. satır: mesajın HTTP POST mesajı olduğunu belirtir. Haberleşme protokolü olarak HTTP kullanılmaktadır.
2. satır: ilgili web servisini sunan sunucuya ait İnternet adresi gösterir.
3. ve 4. satırlar: mesajın “nnnn” uzunluğunda düz metin mesajı olduğunu belirtir.
5. satır: istekte bulunan ilgili yazılım bileşeninin sağladığı servisin spesifik İnternet adresini gösterir.
1. ve 5. satır aralığı SOAP başlık bilgisini içermektedir. 6. ve 14. satır aralığı ise asıl ana kısmı oluşturmaktadır.
7. ve 8. satırlar: genel kurallar ve kodlama kurallarının bulunduğu İnternet adreslerini gösterir.
9. ve 13. satır aralığı içerik kısmını oluşturmaktadır.
10. satır: istekte bulunan yazılım bileşenine ait ilgili metodun adını belirtir.
11. satır: metodun parametre olarak aldığı değeri belirtir.

SOAP başlık bilgisi ve kurallara ait İnternet adres bilgileri kaldırıldığında SOAP istek mesajının düz bir metot çağırma içeriğinden ibaret olduğu kolayca anlaşılabilir. Çünkü SOAP istek mesajı XML standartlarına göre oluşturulduğundan hem bilgisayarlar hem de insanlar tarafından kolayca okunabilmektedir [7].

```

HTTP/1.1 200 OK
Content-Type:text/xml; charset="utf-8"
Content-Length: nnnn

<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  SOAP-ENV:encodingstyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:GetLastTradePriceResponse xmlns:m="Some-URI">
      <Price>34.5</Price>
    </m:GetLastTradePriceResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Şekil 2.7. SOAP cevap mesajı örneği [7]

SOAP cevap mesajı da istek mesajına benzerdir. Mesaj içerisinde geri dönüş değeri olarak ilgili hisse senedine ait fiyat bilgisi yer almaktadır [7].

SOAP, basit ama güçlü bir mekanizma ile bir yazılım bileşeninden başka bir yazılım bileşeninin istenilen metodunun uygun mesajlaşma yapısı sayesinde çağırılabilmesini sağlar. SOAP belirtileri W3C tarafından belirlenmekte ve bütün ana yazılım platformu tedarikçileri tarafından kabul edilerek kendi yazılım yapılarına uygulanmaktadır. Bu sayede farklı yazılım platformları birbirleri arasında SOAP kurallarına göre haberleşebilmektedir [7].

Web Services Description Language (WSDL)

SOAP aracılığı ile yazılım bileşenlerinin birbirlerinin metodlarını çağırabilmeleri için bazı parametrelerin bilinip kullanılabilir olması gerekmektedir. Bu parametreler aşağıdaki gibidir [7].

1. Yazılım bileşenlerinin web servis aracılığı ile kullanıma açtığı fonksiyonlarının bilgileri ve bu fonksiyonların çağrılırken ihtiyaç duydukları parametreler.
2. Haberleşirken kullanılacak bütün XML mesajlarının içereceği veri türleri bilgileri.
3. Haberleşme için kullanılacak protokolün kullanım bilgileri.

4. Servisin sunulduğu ilgili spesifik İnternet adresi bilgileri.

Bu parametrelerin her seferinde el ile otomatik olmadan hazırlanması ve kullanılması özellikle kompleks servislerde zaman alıcı ve hatalara açık olacaktır. Normal şartlarda bu parametrelerin yazılımlar aracılığı ile otomatik olarak hazırlanması ve kullanılması gerekmektedir. Bu amaç doğrultusunda WSDL yapısı geliştirilmiştir [7].

Bu doğrultuda WSDL yukarıda açıklanan parametreleri içeren bir XML dokümanıdır. WSDL dosyalarını işleyebilen yazılım araçları sayesinde parametre işlemleri hızlı ve kolay bir şekilde yapılabilmektedir.

Universal Description, Discovery and Integration (UDDI)

Web servislerinin ve ilgili WSDL dosyalarının istenildiğinde erişilebilmesi ve kullanılabilmesi için bir dağıtım mekanizması gerekmektedir. Özellikle ticari değeri olan uygulamalarda erişilebilirlik, kullanılabilirlik ve performans çok önemli olmaktadır. UDDI sayesinde kullanıma sunulan web servislerinin kayıt edilmesi ve kullanım detaylarının garanti altına alınması sağlanmaktadır. Böylece özellikle ticari uygulamalarda şirketlerin birbirleri arasında kolayca teknik iş modelleri oluşturabilmeleri sağlanmaktadır. UDDI belirtileri sayesinde şirketler birbirleri arasında hızlı, kolay ve dinamik olarak teknik iş süreçleri oluşturup işletebilmektedirler. Temel olarak UDDI şirketlere aşağıdaki faydaları sunmaktadır [7].

1. Sunduğu servisleri tanımlayabilme ve diğer şirketlere kullanım detaylarını iletebilme.
2. Diğer şirketlerin sunduğu servisleri görebilme ve inceleyebilme.
3. Kullanmak istediği servislere kolayca entegrasyonu sağlayabilme.

2.2.2. Web servis teknolojilerinin uygulanması

Genel olarak herhangi bir ağ üzerinden iki farklı yazılım uygulaması haberleşirken aşağıda belirtilen temel koşullar sağlanmak durumundadır [7].

1. Kurulan haberleşme yapısının, uygulamalar arasında sağlıklı bir etkileşimin gerçekleştirilebilmesi için güvenlik, dayanırlılık ve bant genişliği gibi bazı temel performans kriterlerini karşılaması.
2. Yazılım haberleşmeleri ticari bir model aracılığı ile yapılacaksa haberleşmeyi yapacak tarafların ticari parametrelerde anlaşma yapması.
3. Yapılan haberleşme diğer farklı haberleşmelere bağlı ve kompleks ise diğer haberleşmeler ile uyumlu çalışabilmesi.
4. Haberleşme esnasında asenkron iletim gerçekleşmesi veya talep edilmeyen mesajların oluşması durumuna yönelik kayıt ve izleme yapısının olması. Ayrıca bu kayıtlar üzerinde gerekli düzeltme işlemlerin yapılmasını sağlayan mekanizmaların oluşturulması.

Yukarıdaki koşullar web servisleri ortaya atılmadan önce de mevcuttu ve bir çok farklı teknoloji ile bu koşulların sağlanması için çözümler sunulmaktaydı. Web servisleri ortaya atıldıktan sonra bu tür çözüm yaklaşımları sürekli web servislerinin içerisine dahil edilmekte ve gelişmeleri sağlanmaktadır. Bu gelişme süreci içerisinde web servisleri önceleri sadece şirketler içerisinde farklı uygulamalar arasında kullanılmaya başlanmıştır. Zamanla gelişen web servisleri daha sonra sırasıyla önce farklı şirketler arasında uygulanmış arkasından da şirket ve doğrudan müşteri arasında kullanılmasına geçilmiştir [7].

Sonuç itibari ile standart web yapılarının aksine kullanıcı arabirimine sahip olmayan web servisleri, ayrı sunucularda çalışan ve birbirleri arasında veri alışverişi yapan prosesler için program arabirimi sunmakta ve böylece Web üzerinde bilgisayarlardan insanlara bilgi aktarılmasının yanı sıra bilgisayarlardan bilgisayarlara bilgi aktarılıp işlenmesinde de önemli görev üstlenmektedir.

2.3. Anlamsal Web

Web ortamında paylaşılan veri ve bilgiler (düz metin, grafikler, resimler, videolar vb.) insanların anlayıp yorumlayabileceği gösterimlere uygun olarak tasarlanıp oluşturulmaktadır. Bunun sonucunda sunulan bilgiler, farklı yapı ve terminolojide olmasına rağmen insanlar tarafından kolaylıkla anlaşılıp yorumlanabilmektedir. Ancak web ortamındaki bilgi havuzunun giderek genişlemesi, karmaşıklaşması ve dağınıklaşması, bilginin insanların yanı sıra bilgisayarlar tarafından da anlaşılıp yorumlanabilmesi ihtiyaç ve gerekliliğini doğurmuştur.

Bilgilerin, insanların yanı sıra bilgisayarlar ve çeşitli elektronik cihazlar tarafından da yorumlanabilmesi için, yukarıda bahsedilen farklı format ve terminolojinin ortadan kaldırılması ve ortak bir yapının oluşturulması gerekmektedir. Bu doğrultuda Anlamsal Web'i öneren ve gelişmesine ön ayak olan W3C, bilgilerin meta verilerinin belirli standartlar doğrultusunda oluşturulması ve bu standartlar çerçevesinde kullanılıp paylaşılabilmesi stratejisini benimsemiştir. RDF(Resource Description Framework) veri modelleme dili olarak bu amaç doğrultusunda geliştirilmektedir. Ayrıca RDF üzerine kurulmuş olan OWL (Web Ontology Language) ontoloji dili de aynı amaç doğrultusunda geliştirilen bir diğer dildir [36]. RDF ve OWL web ortamındaki veri, bilgi ve kaynakları tanımlama dilleridir.

Bu yaklaşımlar doğrultusunda Anlamsal Web, Web ortamındaki verilerin, bilgilerin kısaca kaynakların bilgisayarların da anlayabileceği şekilde tanımlanması olarak açıklanabilir. Anlamsal Web projesinin amacı, İnternet'teki milyonlarca bilgiyi sınıflandırmaktır. Sistem, bilgileri bağlamlarına göre algılamakta ve aranan bilgiyi bulunduğu ham bağlam ile ilişkilendirmektedir. Sistemde, semantik (anlambilim) metotları kullanılarak veriler birbirleri ile değişik bağlamlarda sayısız kereler yeniden sınıflandırılmaktadır. İnternet'teki milyonlarca veri birbirleriyle o ya da bu şekilde ilişkilendirilebilir; tek bir bilgi birden fazla bağlama dahil olabilir. Elbette, kimi zaman veriler birbirleri ile daha az ilgili olurken, kimi zamansa bağlamın çapı genişletilerek veri sayısı yükseltilebilir.

Bu proje mevcut Web'in, bir uzantısı olarak da görülmektedir. Bu haliyle, verinin web üzerinde hem bilgisayarlar tarafından işlenebilir, hem de insanlar tarafından anlaşılabilir olacak şekilde ortak biçim ile yayınlanmasıdır. Temel fikir, verinin uygulamalar tarafından daha etkili bir şekilde aranabilmesi, işlenebilmesi, yorumlanabilmesi ve entegre edilebilmesidir. Web'in tam potansiyeline ulaşabilmesi için, dünya üzerindeki farklı uygulamaların tam anlamıyla birbirleriyle senkron bir şekilde çalışacakları ve veriyi paylaşabilecekleri bir platform oluşturmak suretiyle, Anlamsal Web'i ortaya çıkarmaları gerekmektedir.

Anlamsal Web, basit noktadan noktaya yapılan bağlantıların ötesinde kişiler, yerler ve kavramlar üzerine kurulu yönlendirmelere olanak sağlarken kullanım esnasında veriyi otomatikleştirme, bütünleştirme ve yeniden kullanma imkanı vermektedir. Bu düşünce aslında, web bilgisinin birleştirilmesi, ilişkilendirilmesi anlamına gelmektedir.

Örneğin, Web ortamında dolaşan bir kullanıcının kendisi ile ilgili bir seminer ilanı görüp seminere katılmaya karar verdiğini düşünelim. İlgili sayfada seminer tarihleri, içeriği, nerede olacağı gibi seminer ile ilgili tüm bilgiler bulunmaktadır. Bu bilgiler kullanıcı tarafından okunarak rahatça anlaşılabilmesine rağmen, kullanıcının bilgisayar bilgileri anlamaz, sadece gösterir. Bu nedenle sayfada bulunan bilgileri kişinin kendisi yorumlamak ve uygun şekilde işlemek durumundadır. Bunun için de seminer tarih bilgilerini alarak takvim uygulamasına girmek, bu tarih aralığında farklı bir planı olup olmadığını kontrol etmek, seminere katılacak diğer kişiler hakkında bilgi edinmek, seminer ücretini ilgili firmanın hesabına yatırmak gibi eylemleri yerine getirmek zorundadır. Hatta kişi ileri teknoloji kullanıyorsa, seminerin yapılacağı yerin adres bilgilerini de GPS cihazına girmek isteyecektir. Tüm bu adımları düşündüğümüzde hepsinin ayrı olarak kişi tarafından gerçekleştirilmesi bir çok açıdan dezavantajdır (Zaman kaybı, iş yükü, hata yapma ihtimali vb.). Bunun yerine kullanıcı, seminer bilgisini ilgili sayfada gördükten sonra sayfanın Anlamsal Web versiyonu mevcut ise yapması gereken tek şey, sayfadaki uygun bölüme "bu seminere katılmak istiyorum" bilgisini girmesi olacaktır. Bundan sonra ilgili web sayfası, geçerli güvenlik kuralları ile birlikte kişinin takvim

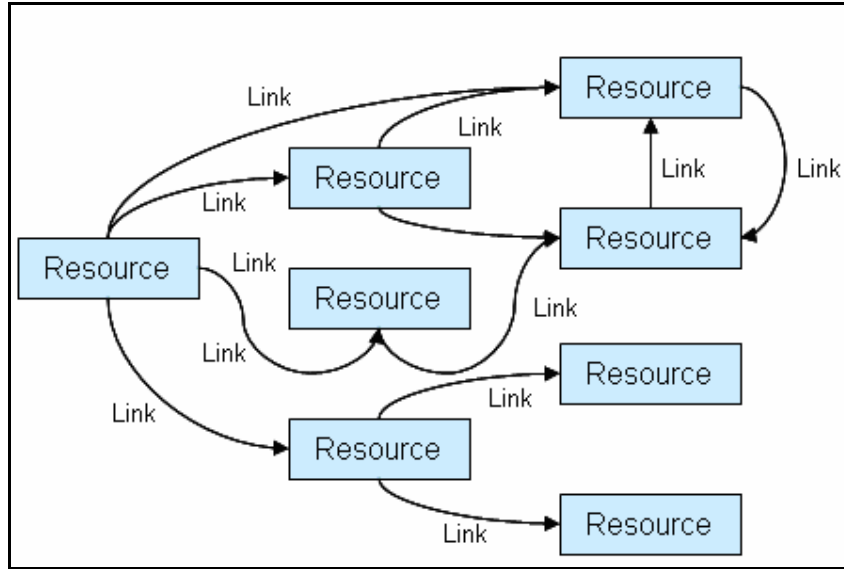
uygulaması ile haberleşip ilgili tarihleri girecek, banka hesabından seminer ücretini çekecek, seminere katılacak diğer kişilerin kişisel web sayfaları Anlamsal Web uyumlu ise, ilgili bilgileri alarak kişiye iletecektir. Hatta gerekirse GPS cihazı ile haberleşerek adres bilgisini cihaza aktaracaktır.

Yukarıdaki örnekten de anlayacağımız gibi bu sistemin hayata geçmesi için en büyük engel, Web içeriğini yaratan kişi, kurum ve kuruluşların bu yapıyı destekleme konusundaki niyetleridir. Anlamsal Web, standart web belgesi yaratma tekniğinden farklı bir yapı kullandığı için şu ana kadar yaratılan sayfaların uyumlu olarak yeniden yazılması gerekecektir.

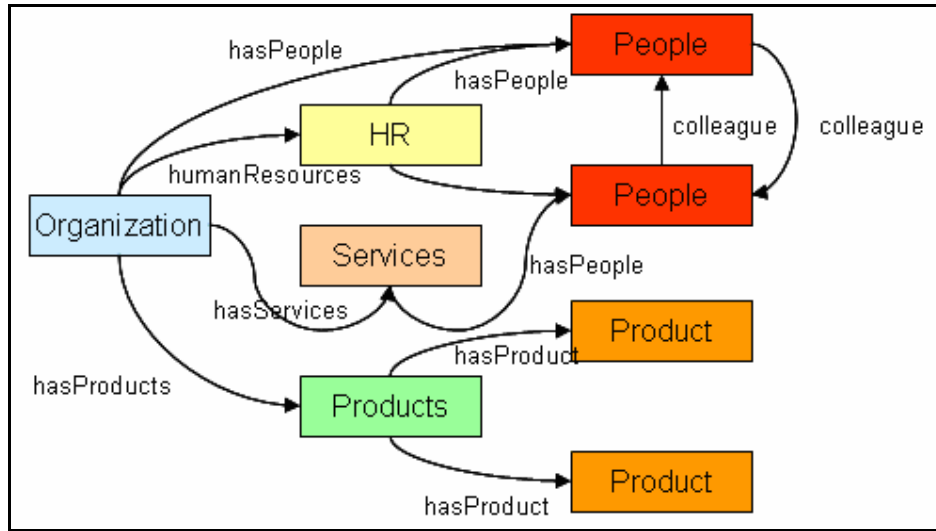
2.3.1. Anlamsal web ve geleneksel web karşılaştırması

Standart web yapısı kaynakların birbirleri arasında düz bağlantılar oluşturulması üzerine kuruludur. Kaynaklar ve bağlantılar arasında anlamsal olarak hiç bir fark bulunmaz. Kaynaklara ve bağlantılara anlam kazandırılması için yeni standartlar ve diller araştırılıp geliştirilmiştir. Bu standartlar ve diller sayesinde kaynaklar için tür belirlemeleri yapıp kaynaklar arasındaki bağlantılara anlam kazandırılabilmiştir. Bu amaçlar doğrultusunda geliştirilen RDF ve OWL standartları sayesinde Web üzerinde bulunan doküman ve verilerin otomatik olarak aranabilmesi ve gerektiğinde yeniden kullanılabilmesi mümkün olmaktadır [29] .

Şekil 2.8’de gösterilen standart web işleyişine göre kaynaklar arasında sadece bağlantılar oluşturulmuştur. Şekil 2.9’da ise kaynaklar tanımlanarak birbirleri arasında oluşturulan bağlantılara uygun anlamlar yüklenmiştir. Bu sayede iyi tanımlanmış ve ilişkilendirilmiş olan bilgilerin ve servislerin Web ortamında bilgisayarlar tarafından okunabilir ve anlaşılabilir olmasını sağlayacak standartlar ve teknolojilerin temeli atılmıştır.



Şekil 2.8. Standart web işleyişi [29]



Şekil 2.9. Anlamsal web işleyişi [29]

Mevcut web uygulamaları genel olarak veri tabanı odaklıdır. Uygulama geliştiriciler uygun veri tabanı modelini oluştururlar ve kullanıcı ile etkileşimde olacak web arabirimlerini bu veri tabanı ile haberleştirerek gerekli işlemlerin yapılmasını sağlarlar. Bu şekilde kurulan yapılar merkezi yapı özelliği göstermekte ve

oluşturulan veri modellerinin kısıtlarından dolayı veri entegrasyonuna olanak sağlayamamaktadırlar [37].

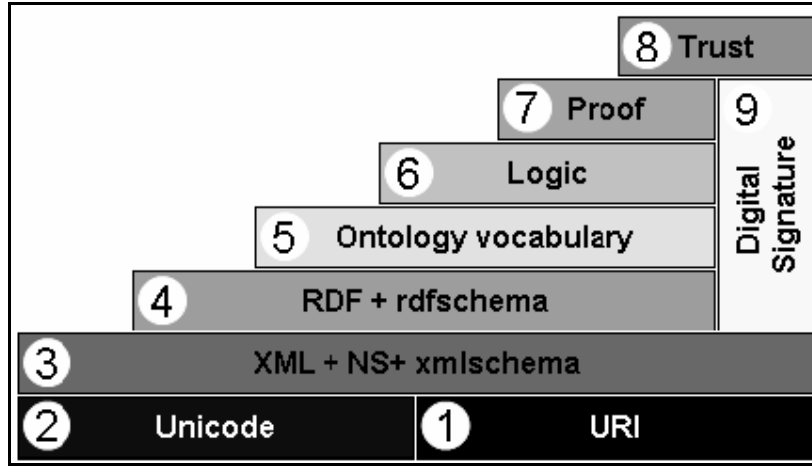
Çizelge 2.2’de temel veri odaklı parametreler doğrultusunda geleneksel web uygulamaları ile Anlamsal Web uygulamalarının karşılaştırılması görülmektedir.

Çizelge 2.2. Geleneksel ve anlamsal web uygulamalarının karşılaştırılması [37]

Geleneksel Web uygulamaları	Anlamsal Web uygulamaları
Merkezi yapı	Dağıtılmış yapı
Belirlenmiş sabit veri modeli	Yarı yapılandırılmış veri modeli
Belirlenmiş sabit veri	İsteğe bağlı veri
Merkezi dağıtım	Her yere dağıtım
Belirli sabit veri kaynağı	Bir çok dağıtılmış veri kaynağı
Kapalı sistem	Açık sistem

2.3.2. Anlamsal web mimarisi

Anlamsal Web, işlevselliğini yerine getirebilmek için alt yapısını oluşturmak üzere bazı teknolojiler, standartlar ve araçlar tanımlar. Anlamsal Web mimarisi birbirleri arasında ilişkiler bulunan bazı standartlaştırılmış yapılardan bütününden oluşmaktadır. Bu mimari genellikle ilk olarak 2001 yılında Tim Berners-Lee tarafından sunulan diyagram ile gösterilmektedir. Aşağıda yer alan Şekil 2.10 Anlamsal Web mimarisinin farklı parçalarını göstermektedir [29].



Şekil 2.10. Anlamsal web mimari modeli [29]

Anlamsal Web mimarisi, URI ve Unicode yapıları ile başlamaktadır. Bu yapıların üzerinde ise söz dizimi geçişini sağlayan XML yapısı bulunmaktadır. Veri modelleme standartlarını oluşturan RDF ve RDF Şema (RDFS) ise XML yapısı üzerine kuruludur. OWL örneğinde olduğu gibi geliştirilen ve web ontoloji dilleri olarak adlandırılan standartlar da RDF ve RDFS üzerine konumlanmaktadır. Son üç katman olan mantık, ispat ve güven henüz tam olarak netleştirilmemiştir. Mimaride yer alan bazı katmanlar, sayısal imza modeline dayanarak güvenlik gerekliliklerini yerine getirirler [29].

Anlamsal Web için temel yapıları ve standartları oluşturan bu katmanlar aşağıda detaylı olarak açıklanmaktadır.

URI ve Unicode

Evrensel Kaynak Tanılayıcısı (URI), soyut veya fiziksel bir kaynağı tanımlamak için kullanılan formatlanmış karakter dizgidir. Bir URI sadece gösterici, sadece isim yada her ikisi birden olabilir. URI'nin bir alt parçası olan İnternet Kaynak Belirtecisi (URL), kaynakları, onlara erişim mekanizmasını göstererek tanımlar. URI'nin diğer bir alt parçası olan İnternet Kaynak Adı (URN) ise genel olarak her bir kaynak için

benzersizdir ve kaynak erişilemez olduğunda dahi varlığını korur. Aşağıda URL ve URN için ilgili örnekler görülebilir [29].

- <http://dme.uma.pt/jcardoso/index.htm> URL'si bir web sayfasının erişilebileceği yeri gösterir.
- <urn:isbn:3-540-24328-3> URN'si ISBN tanımını kullanarak bir kitabı tanımlar.

Unicode; platform, uygulama ve programlama dilinden bağımsız olarak her bir karakter için benzersiz bir sayı ataması sağlar. Unicode yapısından önce birçok çeşit kodlama sistemi mevcuttu. Bu çeşitlilik aynı bilgisayarda birden fazla kodlama sistemi kullanılmasında kodlama çatışmalarına yol açmaktaydı. En eski bilinen kodlama örneği ASCII'dir [29].

Extensible Mark-up Language (XML)

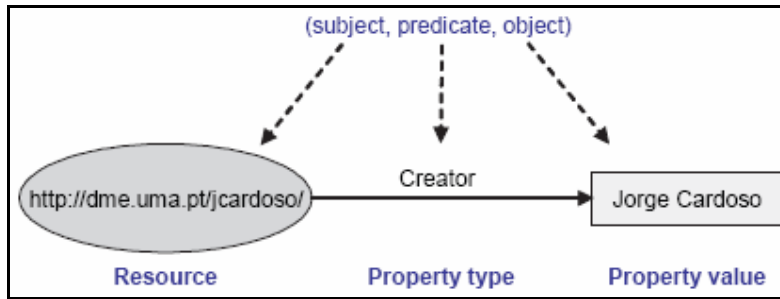
Anlamlarından bağımsız olarak verilerin yapısal olarak formatlanması standardı olarak kabul edilmektedir [29]. Daha çok Web ortamında kullanılmaktadır. Meta veri kullanımı ile verilerin gösteriminde de bir standart sunan ve aynı zamanda esnek kodlamaya izin veren XML veri entegrasyonları için en çok önerilen ve kullanılan çözümdür.

Dünya genelinde yoğun kullanımı nedeni ile XML'i veri formatlama için önemli bir çözüm olarak kabul etmek gerekir. Ancak anlamsal entegrasyonlar açısından düşünüldüğünde XML'in bazı kısıtlamaları vardır. Bunlardan en önemlisi, XML verinin yapısal standardına çözüm getirmesine rağmen belirli domainlerin anlamlarının tanımlaması için herhangi bir çözüm sunmamaktadır. Ayrıca anlamsal yapılar ve ilişkiler düşünüldüğünde XML'in veri modelinin zayıf kaldığı görülmektedir [29].

Resource Description Framework (RDF)

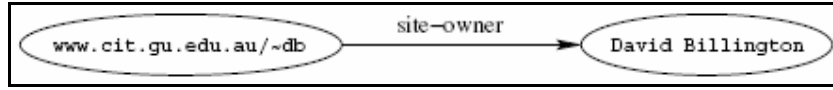
Meta veri tanımlamalarının ve kullanımlarının standart hale getirilmesi için W3C tarafından geliştirilen bir dil olan RDF, katmanlı modele göre XML yapısının üzerine kuruludur. RDF çalışma mekanizmasında XML'i kullanır ve bu Anlamsal Web'in temelidir. Katmanlı modele göre geliştirilen diğer tüm ontoloji dilleri bu temel üzerinde yer alır. RDF, web kaynaklarının tanımlanmasında kullanılan ve bilgisayarlar tarafından anlaşılabilir şekilde meta veri standartları içeren biçimsel bir veri modelidir. XML uygulamaları için standartlar oluşturan RDF sayesinde entegrasyon ve haberleşme yöntemleri kolaylaşmıştır [29].

RDF veri modeli soyut olarak nesneye yönelik modele benzemektedir. Model, genel olarak benzersiz göstericiler tarafından tanımlanan varlıklar kümesinden oluşmaktadır. Ayrıca bu varlıklar arasında ikili ilişkiler ve durumlar bulunmaktadır. RDF deyimlerinin grafiksel gösterimlerinde ilişkinin kaynağına özne, iki düğümü birleştiren öğeye yüklem (veya özellik) ve ilişkinin hedefine de nesne denilmektedir [38].

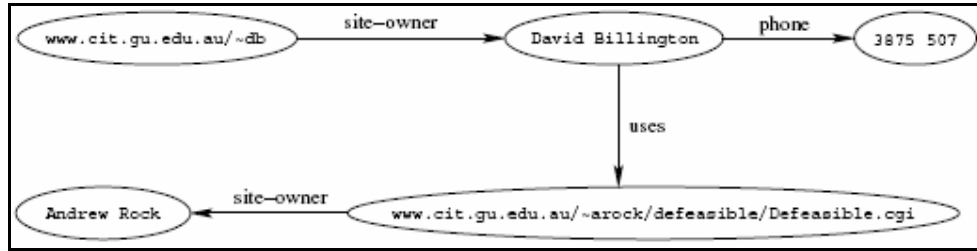


Şekil 2.11. RDF deyimini grafik gösterimi [29]

Bir ilişkideki herhangi bir nesne farklı bir ilişki için özne (kaynak) olabilir. Şekil 2.12'deki RDF deyiminde nesne olarak görülen "*David Billington*", Şekil 2.13'deki RDF deyiminde özne olarak yer almaktadır.



Şekil 2.12. RDF deyimi grafik gösterim örneği [39]



Şekil 2.13. RDF çoklu deyim grafik gösterim örneği [39]

Şekil 2.12'deki RDF deyimine ait XML tabanlı RDF gösterimi aşağıda görülmektedir. XML standartlarına ek olarak RDF yapısına özgü etiketler kullanılarak uygun RDF dokümanı oluşturulur.

```
<?xml version="1.0" encoding="UTF-16"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:mydomain="http://www.mydomain.org/my-rdf-ns">

  <rdf:Description rdf:about="http://www.cit.gu.edu.au/~db">
    <mydomain:site-owner>
      David Billington
    </mydomain:site-owner>
  </rdf:Description>

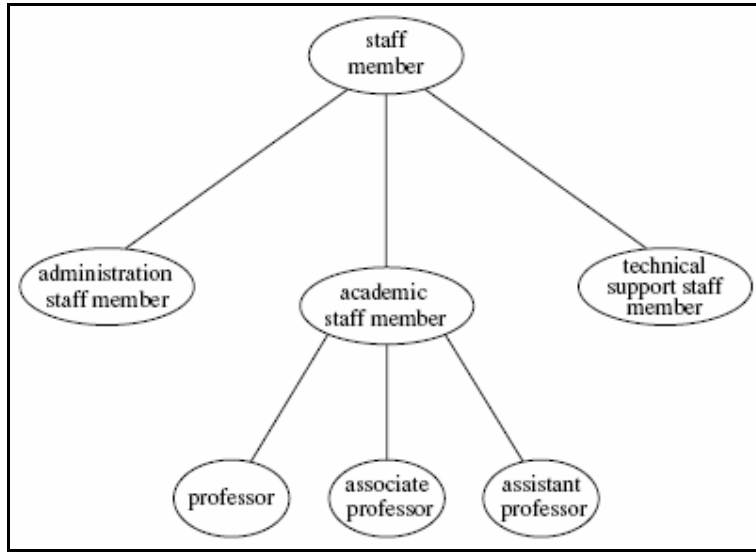
</rdf:RDF>
```

Şekil 2.14. RDF deyimi XML tabanlı gösterimi [39]

Birinci satır XML kullanıldığını belirtir. İkinci satırdaki “*rdf:RDF*” etiketi oluşturulan dokümanın RDF dokümanı olduğunu açıklar. Üçüncü satırdaki “*rdf:Description*” elemanı deyimdeki özne kaynağını tanımlar [39].

RDF Schema (RDFS)

RDF şema, genel olarak RDF için bir, tür sistemi sağlamaktadır. RDF şema ile kaynaklar; sınıf, özellik ve değerler ile birlikte tanımlanabilmektedir. RDF şemadaki sınıf özellikleri nesneye yönelik programlama dillerinde olduğu gibidir. Bir sınıf benzer varlıkların yapısal oluşumudur ve bu oluşum miras özelliklerine izin vermektedir [29].



Şekil 2.15. RDFS için sınıf hiyerarşisi örneği [39]

RDF, W3C tarafından XML dili üzerine kurulu olacak şekilde verinin özelliklerini ve anlamlarını belirten bir standart olarak geliştirilmiştir. Buradaki amaç, standart bir biçimde, bilginin özellik ve sınıfı belirtebilmek için bir veri modeli sağlayarak, web bilgisine anlam katabilmektir. Bu sayede RDF, veri kaynakları arasındaki ilişkileri belirleyerek verilerin anlamını açıklayabilmektedir. RDF çalışma grubu tarafından, en küçük ontoloji modelleme dili olarak düşünülebilecek nesne yönelimli yapıdaki RDF Schema (RDFS) geliştirilmiştir.

RDF ve RDFS Anlamsal Web alanında veri modelleme için bir başlangıç oluştursa da bir çok konuda yetersiz kalmaktadırlar. Örneğin özelliklerin özelliklerinin

tanımlanamaması veya sınıfların alt sınıflarının oluşturulamaması gibi eksiklikler, gerçek yaşamdaki veriler arası ilişkileri tam anlamıyla ifade edemezler. Bu sebeple ontoloji kurmak için, gerçek yaşamdaki kavramsal olarak bilgileri ifade edebilecek ontoloji dilleri geliştirilmiştir.

Ontoloji

Ontoloji, kelime olarak Türkçe'ye "Varlık Bilimi" olarak çevrilmektedir. Varoluşu araştıran felsefedir. Bilgisayar dünyasındaki çalışmalarda ontoloji, veriyi işlenebilecek bir varlık halinde algılamak olarak değerlendirilmektedir. Bilgi, tek başına bir varlık olarak düşünölemeyeceğı için, bilgisayarların bilgiyi işlemesi amacıyla, onu varlık haline dönüştürmesi, yani bilgiye ait özellikler ve özellikler arası ilişkiler tanımlaması gerekmektedir. Böylelikle bilgi, kendisine ait özellikler ile sınıflandırılabilcek, diğler bilgilerle ilişkilendirilebilecektir.

Ontoloji, bir alana ait bilgiyi tanımlayabilmek ve açıklayabilmek için gerekli terimleri tanımlar. Buradaki alan herhangi bir konuda olabilir, otomotiv, sağlık, endüstri vb. Ontolojiler, oluşturuldukları alanlara ait bilginin, insanlar, veritabanları ve uygulamalar tarafından paylaşılabilmesini sağlarlar. Alanların içerisindeki bilgileri, ve alanları birbirinden ayıran bilgileri ayrıştırırlar.

Bir ontoloji, çalıştığı alan içerisindeki varlıkları, birbirleriyle olan ilişkileri üzerinden tanımlar ve isimlendirir. Ontoloji, bir alan içerisindeki bilgiyi ve alan içerisindeki ilişkileri sunabilmek ve o bilgiyle kullanıcılar arasındaki iletişimi kurabilmek için bir sözlük oluşturur. Bir ontoloji, bir alana temel oluşturan veritabanı, dosya sistemi, veri dosyaları vs. yapısal veya yarı-yapısal verilerin mantıksal gösterimini sağlar. Bir başka deyişle ontoloji, bir kavramı sözdizimi olarak tanımlamak yerine, ifade ettiğı ve taşıdığı anlam bağlamında açıklar. Bu ayırım ile farklı veri kümeleri arasında kesin bir farklılık ortaya konulabilir. Böylelikle birbirine çok yakın anlam taşıyan veriler, yahut kelime olarak aynı olan veriler birbirinden ayrıştırılır. Mesela bir veritabanında alan ismi olarak "WAGE" veya "SALARY" kullanılabilir. Anlam olarak birbirine

çok yakın olan bu iki kelime arasında anlamsal olarak küçük ama önemli farklılık vardır. Veritabanını desenine göre işlem yapan bir yazılım, bu iki alanı aynı görebilir. Ancak ontoloji bu iki kavramın aynı olmadığını ortaya koymaktadır. Bu yüzden ontoloji kullanan bir yazılım bu kavramların ayrı anlam taşıdıklarını bilir.

Bilgi sistemlerindeki ilk ontolojiler bilgi tabanlarında kullanılmıştır. Anlamsal Web'in çıkışı ile birlikte de web ontolojileri önem kazanmıştır. Bir web ontolojisi için belirlenen gereksinimler doğrultusunda OWL dili paylaşılabilen, evrimleşen ve bir arada çalışabilen ontolojileri destekleyebilecek şekilde tasarlanmıştır. OWL dili ile yaratılmış bir web ontolojisi sınıflardan, bu sınıflara ait niteliklerden ve bu sınıfların örneklerinden oluşur. Ayrıca bu sınıflar arasındaki hiyerarşik ve mantıksal ilişkiler, sınıf niteliklerine ait niteliklere uygulanabilecek değer kısıtlamaları ve mantıksal kısıtlamalar da OWL ontolojisine dahil edilebilir.

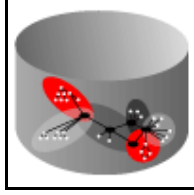
OWL dili günümüz web teknolojilerinin üzerine inşa edilmiştir. XML sözdiziminde olduğu ve XML ve RDF isim uzayından elemanlar kullandığı için varolan uygulamalar OWL dili ile yazılmış bir dokümanı, içerdiği bilgiyi anlayamamalarına rağmen, çözümleyebilirler. Bu sayede varolan yazılım bileşenleri değişikliklerle OWL dilini destekler bir hale getirilebilir.

OWL dışında en çok kullanılan ontoloji dilleri DAML+OIL ve SHOE'dir. SHOE'in geliştirilmesi sonlansa da, halen geçerliliğini korumaktadır.

2.3.3. Anlamsal web uygulama örnekleri

Anlamsal Web henüz gelişim sürecinde olmasına rağmen şimdiden bir çok kişi ve kurum tarafından uygulama ve geliştirme araçları oluşturulmuş ve Anlamsal Web sistemlerinin kurulması için çalışılmaktadır [29]. Özellikle yazılım platformları için alt yapı sağlayan kuruluşlar gerekli entegrasyonları yapmaktadırlar.

Oracle firması, Anlamsal Teknolojiler Merkezi isminde bir birim kurarak Anlamsal Web ile ilgili çalışmalarına başlamıştır [40].



Şekil 2.16. Oracle, anlamsal teknolojiler merkezi [40]

Adobe firması, Acrobat gibi ürünlerine Anlamsal Web özelliğini kazandırmaya başlamıştır. Bunun için XMP (Extensible Metadata Platform) teknolojisini geliştirmiştir [41].



Şekil 2.17. Adobe, extensible metadata platform [41]

IBM firması, Integrated Ontology Development Toolkit ürününü çıkarıp Anlamsal Web alanındaki geliştirmeler için destek vermektedir [42]. HP firması, Semantic Web Research Labs isminde birim kurarak Anlamsal Web ile ilgili çalışmalar yapmaktadır [43].

Anlamsal Web alanındaki uygulamalara bir diğer önemli örnek de web arama motorlarıdır. Günümüzün lider arama motorları (Google, Yahoo, vb.) sadece anahtar kelime, meta arama, indeksleme ya da buna benzer ilk-nesil teknolojiler kullanmaktadır. Diğer taraftan geliştirilen anlamsal arama motorları sorulan soruları algılama ve komple cevap verme ve otomatik olarak öğrenme becerileri gibi özelliklere sahiptirler. Microsoft firmasının geliştirdiği Miss Dewey [44], Hakia firmasının geliştirdiği hakia [45] ve Maryland Baltimore Country Üniversitesinde geliştirilen Swoogle [46] bunun için bazı örneklerdir.



Şekil 2.18. Hakia, anlamsal arama motoru [45]

2.3.4. Anlamsal web yazılım araçları

Anlamsal web kapsamında geliştirilecek uygulamaların kolaylıkla oluşturulabilmeleri için ortaya çıkarılmış bir çok yazılım aracı mevcuttur. Bu yazılım araçlarının daha yetenekli olabilmeleri yönünde her geçen gün geliştirilmelerine devam edilmektedir. Anlamsal web veri modellerinin otomatik yöntemlerle oluşturulup kullanılabilmelerine olanak sağlayan bu araçlar sayesinde özel uygulamalar çok daha kolay geliştirilebilmektedir.

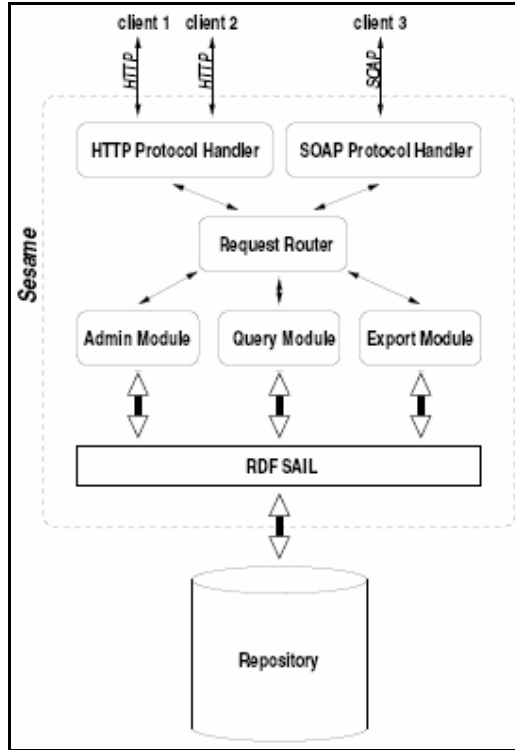
Sesame ve Protege en çok kullanılan anlamsal web yazılım araçlarıdır. Ayrıca Sesame ve Protege arasında veri ve veri modellerinin iletilebilmesini sağlayan entegrasyon yazılımları da mevcuttur.

Sesame

Sesame [47], RDF ve RDF Şema ile oluşturulmuş veri modellerinin depolanması ve uygun şekilde sorgulanması için geliştirilmiş ve içerisinde çeşitli görevleri yerine getiren modüllerden oluşan bir platformdur [48]. Depolama işlemi için üç farklı yöntem kullanılabilir [49].

1. Veri modelinin bellek içerisine oluşturulması ve bellek üzerinden kullanılması. Gerektiği durumlarda kalıcılığın sağlanması için sabit disk üzerinde uygun dosya içerisine yedekler alınabilmektedir.
2. Veri modelinin sabit disk üzerinde oluşturulacak özelleştirilmiş dosya sistemi içerisinde oluşturulması ve bu dosya sistemi üzerinden kullanılması.
3. Veri modelinin desteklenen bir ilişkisel veritabanı içerisinde oluşturulması ve bu veritabanı üzerinden kullanılması.

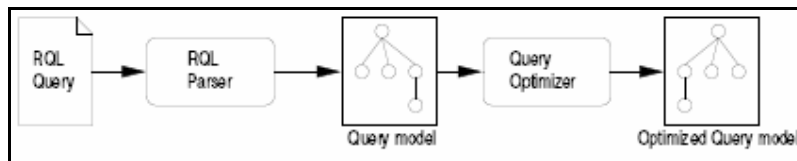
Sesame mimarisi, farklı görevleri yerine getiren ve birbirleri ile entegre bir şekilde çalışan çeşitli modüllerden oluşmaktadır. Bu modüllerin en önemlisi istemcilerden anlamsal web veri modelleri kapsamında gelecek istekleri yorumlayıp farklı alternatiflerde oluşturulabilecek depolama sistemi üzerinde gelen istekler doğrultusunda gerekli işlemleri yürüten ve SAIL (Storage And Inference Layer) olarak adlandırılan bir uygulama programı arayüzüdür [48].



Şekil 2.19. Sesame mimari yapısı [48]

İstemcilerin sesame platformu ile iletişim kurması için HTTP, SOAP ve RMI gibi bir çok farklı arabirim bulunmaktadır. İstemciler bu farklı arabirimlerden kendilerine uygun olanı seçip gerekli belirtileri yerine getirerek sesame platformu ile haberleşebilmektedirler [48].

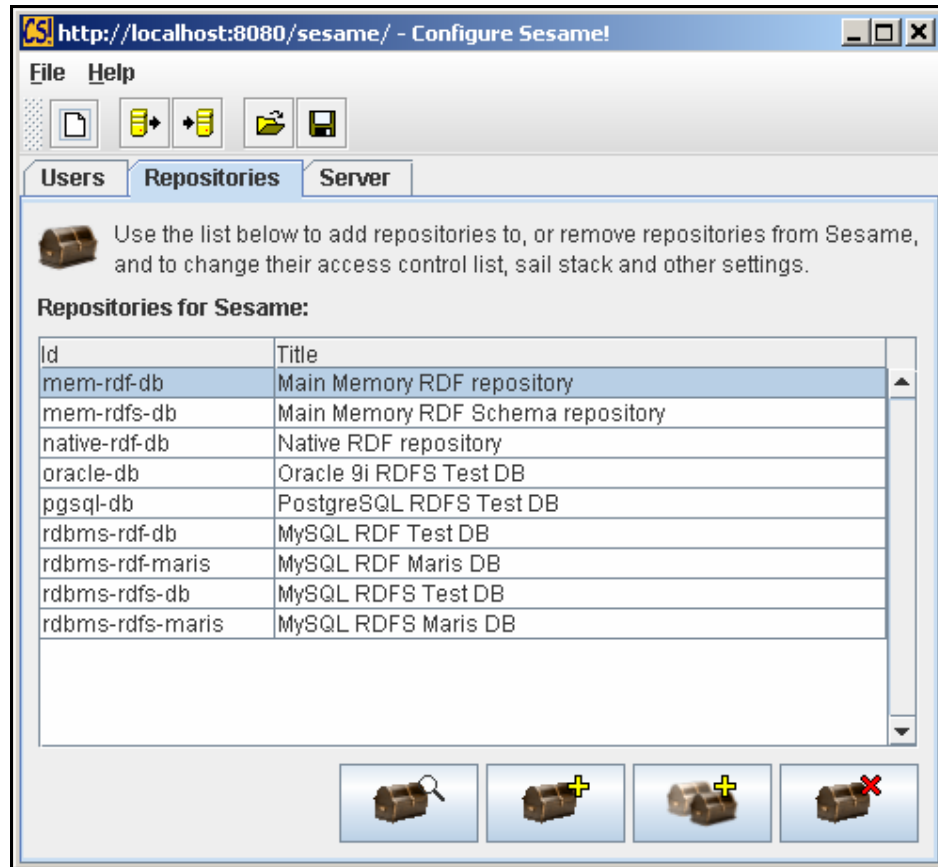
Sesame platformu içerisinde yer alan bir diğer önemli modül RDF sorgulama modülüdür. Bu modül sorgulama yönetimini Şekil 2.20’de gözüktüğü gibi yapmaktadır [48].



Şekil 2.20. Sesame sorgu çözümleme ve optimize etme modeli [48]

Sesame platformu üzerinde gerekli ayarlamaları ve yapılandırmaları gerçekleştirmek için gelişmiş bir kullanıcı arabirimi sunulmaktadır. Bu arabirim aracılığı ile gerçekleştirilebilecek temel fonksiyonlar aşağıdadır.

- Sesame platformu üzerinde işlemler yapacak kullanıcılar ve bu kullanıcılar için uygun hakların tanımlanması.
- RDF ve RDFS veri modellerinin yaratılması ve bu veri modellerine ilişkin uygun parametrelerin yapılandırılması.
- Sesame sunucuna ilişkin yönetici erişim parametreleri ve kayıt yapma parametreleri gibi özelliklerin yapılandırılması.

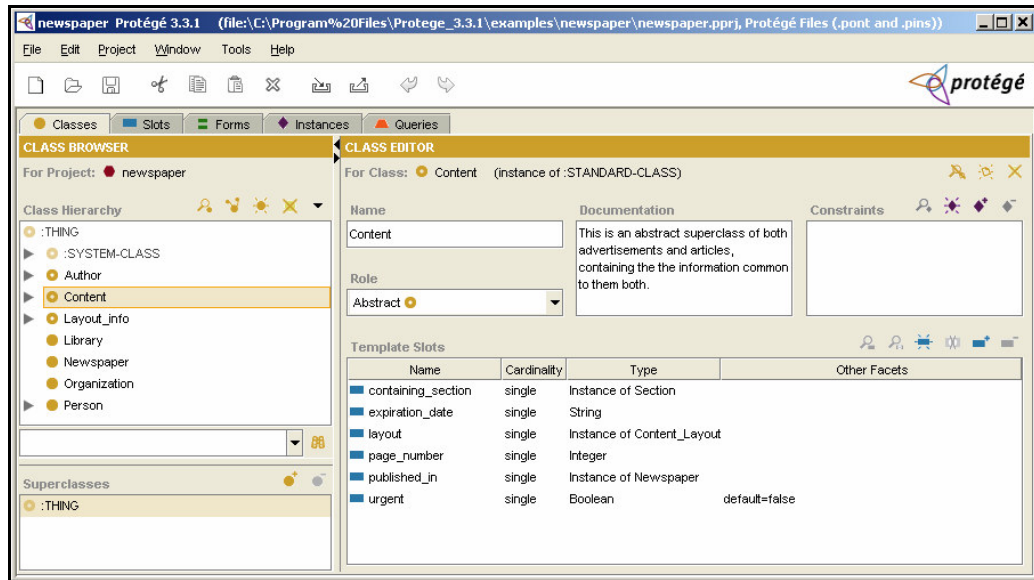


Şekil 2.21. Sesame yapılandırma arabirimi [48]

Protege

Protege [50], anlamsal web veri modelleri için yüksek oranda uyarlanabilir yeteneklere sahip bir belge düzenleyicidir. Uyarlanabilirliği sağlayan temel özellikler aşağıda açıklanmaktadır [51].

- *Geliştirilebilir bilgi modeli*; Tanımlanmış sistem öğelerinin kullanıcı tarafından uyarlanabilir şekilde yeniden tanımlanabilmesi.
- *Uyarlanabilir dosya yapıları*; Anlamsal web veri modellerine göre üretilen yapıların farklı dosya formatlarında oluşturulup kayıt edilebilmesi.
- *Uyarlanabilir kullanıcı arabirimi*; Kullanıcı arabiriminin sunduğu seçeneklerin kullanıcı tarafından farklı alternatiflere göre yapılandırılabilmesi.
- *Diğer uygulamalar ile entegrasyon için uyarlanabilir mimari*; Oluşturulan veri modellerinin diğer anlamsal web uygulamaları ve yazılım araçları ile kolay bir şekilde entegre edilebilmesi.



Şekil 2.22. Protege kullanıcı arabirimi

3. BAĞLAM BİLİNÇLİ UYGULAMALAR

Günümüzde bilgisayar uygulamaları geleneksel masaüstü uygulamalarından mobil uygulamalara doğru kaymaktadır. Bunun en önemli sebebi kullanıcıların hareketlilik parametrelerini ve dinamik çevresel koşullarını işleyebilecek şekilde gelişen mobil cihazlardır. Bunun sonucu olarak kullanıcılara ait dinamik bağlam bilgilerinin işlenebilmesi ve kullanıcılara çok daha fazla fayda sağlayabilecek imkanların sunulabilmesi ortaya çıkmıştır [52].

Geleneksel bilgisayar uygulamaları Şekil 3.1’de gözüktüğü gibi işlem yapmaktadırlar. Uygulama girişleri açık ve net verilerden oluşmaktadır. Bu çalışma sisteminde dinamik çevre koşullarından ve kullanıcıya ait değişken parametrelerden yararlanılamamaktadır [52].

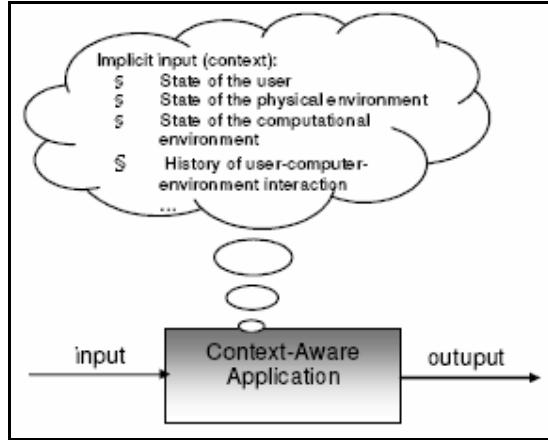


Şekil 3.1. Geleneksel bilgisayar uygulamaları [52]

Bağlam bilinçli uygulamalar kullanıcılara gelişmiş servisler sunabilmek için bağlamsal bilgileri işleyebilen bilgisayar sistemleri oluşturmayı amaçlamaktadır. Bunun için kullanıcı ile uygulama arasındaki etkileşimini artırarak mobil parametrelerin uygun şekilde kullanılmasına yönelirler. Bağlamsal bilgiler, özellikle kullanıcının hangi zamanda nerede ne yaptığı ile ilgilidir. Kullanıcının işlenen konular üzerine bilgileri ve ayrıca sistemin bilgileri uygulamanın performansını artırıcı değerlerdir. Bu parametreler doğrultusunda ortaya çıkan sonuçların kullanıcının faydasına yönelik olması bağlam bilinçli uygulamalar için en önemli amaçtır.

Bağlam bilinçli uygulamalar Şekil 3.2’de gözüktüğü gibi işlem yapmaktadır. Uygulamalar, giriş olarak alınan açık verilere ek olarak dinamik bağlamsal bilgileri

de dikkate almaktadırlar. Bu dinamik bağlamsal bilgiler çoğunlukla uygulamaya ait çevresel parametrelerden toplanmaktadır [52].



Şekil 3.2. İçerik bilinçli uygulamalar [52]

Genel olarak bağlam bilinçli bir uygulama aşağıdaki özelliklere sahip olmalıdır [17].

1. Bilgi ve servislerin uygun arabirimlerle kullanıcıya sunulması.
2. Kullanıcıya sunulan bilgi ve servislerin otomatik olarak işletilmesi.
3. Kullanılan içeriğin standartlara uygun olarak işaretlenip sonraki kullanımlar için bilgi olarak saklanması.

3.1. İçerik ve Bağlam

Filtrelenmiş içerik veya bağlam için literatürde değişik amaçlar doğrultusunda bir çok farklı tanım yapılmıştır. Her ne kadar kavramsal olarak insanların kafasında fikir oluşsa da net bir tanım yapmak zor olmaktadır. Farklı açılardan bakıldığında farklı açıklamalar yapılması mümkündür. Burada yaygın bilişim penceresinden bakılarak bir kaç değişik tanım verilecektir [52].

1994 yılında Schilit ve Theimer [16] tarafından ortaya atılan düşünceye göre bağlam için önemli olan parametreler; kullanıcının yeri ve yakınında bulunan diğer

insanlardır [52]. Ayrıca Schilit bağlam tanımlamasını farklı üç kategoriye bölmüştür. Bunlar [18];

1. *Bilgi işleme bağlamı*; ağ bağlantıları, haberleşme maliyetleri ve haberleşme bant genişliği. Ayrıca yazıcı, görüntü arabirimleri ve iş istasyonları gibi yakında bulunan kaynaklardır.
2. *Kullanıcı bağlamı*; kullanıcı profili, bulunduğu yer, yakınında bulunan diğer insanlar ve hatta o anda içinde bulunduğu ruh hali gibi parametrelerdir.
3. *Fiziksel bağlam*; ışık durumu, gürültü seviyesi, trafik şartları ve sıcaklık gibi parametrelerdir.

Bunların dışında zaman bilgisi de ayrıca önemlidir ve bir çok uygulama için doğal bağlamdır. Zaman bilgisini yukarıdaki kategorilerden herhangi birine koymak zor olduğu için dördüncü bir kategori olarak ayırmak uygun olacaktır [18].

4. *Zaman bağlamı*; o andaki zaman, ilgili hafta, ay ve ilgili yıla ait mevsim gibi parametrelerdir.

Ayrıca diğer bir önemli nokta da bağlam kategorileri ile ilgili işlemler yapılırken ilgili bağlam bilgilerinin geçmiş dönemli kayıtlarının tutularak bağlam tarihçesinin oluşturulması bir çok uygulama için faydalı olacaktır [18].

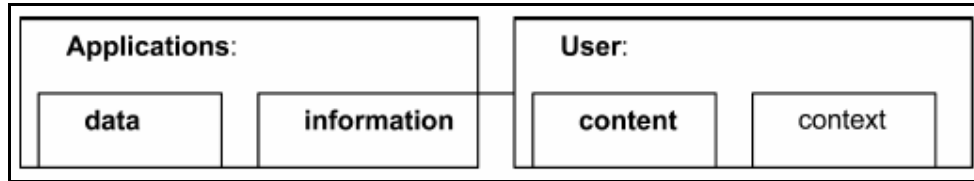
Bunun dışında Pascoe [53] bağlamı; özel bir varlığın fiziksel ve kavramsal durumlarını içeren alt kümeler olarak tanımlamıştır. Bu tanım daha çok operasyonel anlam içermektedir ve çok fazla spesifiktir. Bağlam daha çok herhangi bir uygulamanın ve uygulama kullanıcılarının bütün durumları ile ilişkilidir. Parametreler duruma göre değişeceği için her durumlarda statik olarak özellikleri sıralamak doğru olmayacaktır. Bu nedenle Dey [17] bağlamı aşağıdaki gibi tanımlamaktadır;

Bağlam, bir varlığın durumunu karakterize edebilmek için kullanılan herhangi bir bilgidir. Bu varlık; bir insan, bir yer veya uygulama ve kullanıcı dahil olmak üzere

uygulama ile kullanıcı etkileşimiyle ilgili olan herhangi bir nesne olarak düşünülebilir.

Bu tanım sayesinde uygulama geliştiriciler için herhangi bir uygulama senaryosunda ilgili bağlamları sıralamak kolay olacaktır. Eğer, kullanıcının uygulama ile etkileşimde herhangi bir bilgi herhangi bir durumu karakterize etmede kullanılıyorsa bu bilgi bağlamdır [17].

İçerik ve bağlam arasındaki fark en iyi şekilde uygulamalardaki veri ve bilgi arasındaki fark benzerliği ile ifade edilebilir. Bu anlamda içerik, kullanıcının eriştiği ve belirli amaca yönelik işlemlerde kullanılan veri içerisindeki bilgi değeridir. Diğer taraftan bağlam ise kullanıcının anlık ilgi alanları, bulunduğu yer, zaman ve kişinin bakış açısına göre ileri düzeyde haberleşme gereklilikleri ve özellikleri için daha çok kullanıcı ile ilişkili olarak işlenilen bilgidir. Bu parametreler doğrultusunda kullanıcı çok daha spesifik ve gerçek zamanlı bilgiye erişiyor olmaktadır [54].



Şekil 3.3. Uygulama ve kullanıcı açısından bilgi erişimi karşılaştırması [54]

3.2. Bağlam Farkındalığı

Bağlam farkındalığı, kullanıcı çevresel etkileri gibi bağlam bilgilerini işleyebilen bilgisayar birimleri oluşturarak insan bilgisayar etkileşimini mümkün olduğunca açığa çıkarmayı amaçlamaktadır. Bağlamın bilinçli olması, kullanıcının bilgisayar sistemine girdi olarak el ile ileteceği açık ve net parametreleri azaltmak için kullanılmaktadır [52].

Mobil uygulamalar için kullanılabilecek bağlam bilgileri günümüzde mevcut olsa da bu bilgilerin en etkin şekilde nasıl kullanılabileceği uygulama geliştiricileri açısından hala tartışılması gereken bir konudur. Bu kapsamda Schilit, bağlam bilinçli bilgi işlemeyi, bağlam bilinçli uygulamaları aşağıdaki gibi sınıflandırarak tanımlamaktadır [18].

1. *Yaklaşık seçme*; yakın nesnelerin kullanım için seçilmelerine dayanan bir kullanıcı arabirimi oluşturma tekniğidir.
2. *Otomatik bağlamsal yeniden yapılandırma*; yeni bileşenlerin eklenmesi, var olanların çıkartılması veya bağlamdaki değişimlere göre bileşenlere olan bağlantıların güncellenmesi tekniklerine dayanan bir işleme yöntemidir.
3. *Baglamsal bilgi ve yorumlama*; ortaya çıkarılan bağlama göre farklı sonuçlar üretilmesidir.
4. *Bağlama dayalı tetikleme*; bağlam bilinçli sistemin adaptasyonunu belirlemek için kullanılan geleneksel IF-THEN kurallarıdır.

3.3. Mobil Cihazlarda Bağlam Bilinçli Uygulamalar

Mobil İnternet kavramı genel olarak üç farklı noktadan değerlendirilerek anlaşılabilir [11]. Bunlar;

1. *Kullanıcı açısından*; Mobil İnternet özelliğine sahip cihazlar genellikle sabit İnternet cihazlarına göre çok daha fazla kişiselleştirilmiştir. Kullanıcıya ait özelliklere göre ayarlanabilen mobil cihazlar her zaman için kişinin karakter bilgilerini taşımaktadır.
2. *Çevresel etkiler açısından*; Mobil İnternet sistemleri genellikle anlık olarak İnternet'e erişim sağlayabilirler. Böylece kullanıcı istediği her yerden istediği her an İnternet'e ulaşabilir. Bunun yanında sabit İnternet sistemleri ise çoğunlukla hareketli olabilme özelliklerinden yoksundurlar ve kullanıcının erişimi gerektiğinde zaman harcanan bazı işlemlerin yapılması gerekir (Erişim için kullanılacak PC'nin açılması için geçen süre vb.).

3. *Mobil İnternet sistemleri açısından;* Özellikle cep telefonu sistemleri başta olmak üzere çoğu Mobil İnternet sistemleri sabit İnternet sistemlerine kıyasla kullanıcıya çok daha az kaynak sağlayabilmektedir. Mobil İnternet cihazları çok pratik ve kolay kullanışlı olmaları yanında diğer taraftan da daha küçük ekranlara, giriş çıkış işlemleri için daha az imkanlara ve daha az bilgi işleme kapasitelerine sahiptirler.

Mobil İnternet sistemlerinde kullanıcılar açısından yazılım arabirimleri düşünüldüğünde sabit İnternet sistemlerine benzer şekilde iki farklı yapının mevcut olduğu görülür. Bunlar;

1. *Tarayıcı tabanlı arabirimler;* Mobil cihazlarda mevcut olan tarayıcı yazılımları Mobil İnternet erişimi için arabirim olarak kullanılırlar. Kullanıcı, tarayıcı yazılımını çalıştırır ve erişmek istediği web sayfasının adresini girerek istediği sayfaya ulaşır. Telephia firmasının [13] 14 Ağustos 2006 tarihli raporuna [55] göre Amerika'daki Mobil İnternet tarayıcı kullanıcılarının en çok tercih ettikleri tarayıcı yazılımları sırasıyla; Openwave [56], Motorola [57] ve Nokia [58]'dir. Mobil İnternet sistemleri için geliştirilen web sayfaları çoğunlukla WML [59] işaretleme diline göre yapılandırılırlar.
2. *Uygulama tabanlı arabirimler;* Mobil İnternet sistemleri için özel olarak geliştirilen uygulamalar mobil cihazlara yüklenir. Kullanıcı, cihazda yer alan uygulamayı çalıştırarak kendisine sunulan arabirim üzerinden işlemlerini yerine getirir. Mobil cihazlar için uygulama geliştirmeye yönelik bilinen iki önemli platform; SUN firmasının sunduğu J2ME (Java 2 Micro Edition) [60] ve Microsoft firmasının sunduğu .NET Mobile [61]'dir [15].

Telephia firmasının 14 Ağustos 2006 tarihli raporuna [55] göre Amerika'da elektronik posta ve hava durumu sayfaları mobil İnternet kullanıcılarının en çok eriştikleri iki ana mobil İnternet içerik kategorisidir. Bunu spor ve arama kategorileri takip etmektedir. 2006 yılının Haziran ayına ait detaylı kategori sıralaması aşağıdaki çizelgede sunulmuştur.

Çizelge 3.1. En çok ziyaret edilen mobil web sayfaları (Haziran 2006) [55]

Mobile Website	Unique Audience (000)	Reach of Subscribers
1. Yahoo! Mail	6,531	3.0%
2. The Weather Channel (Weather.com)	5,827	2.7%
3. ESPN	5,345	2.5%
4. Google Search	4,356	2.0%
5. MSN Hotmail	3,441	1.6%
6. MapQuest	3,067	1.4%
7. AOL Mail	2,907	1.4%
8. CNN	2,799	1.3%
9. Yahoo! Weather	2,740	1.3%
10. Yahoo! Search	2,531	1.2%

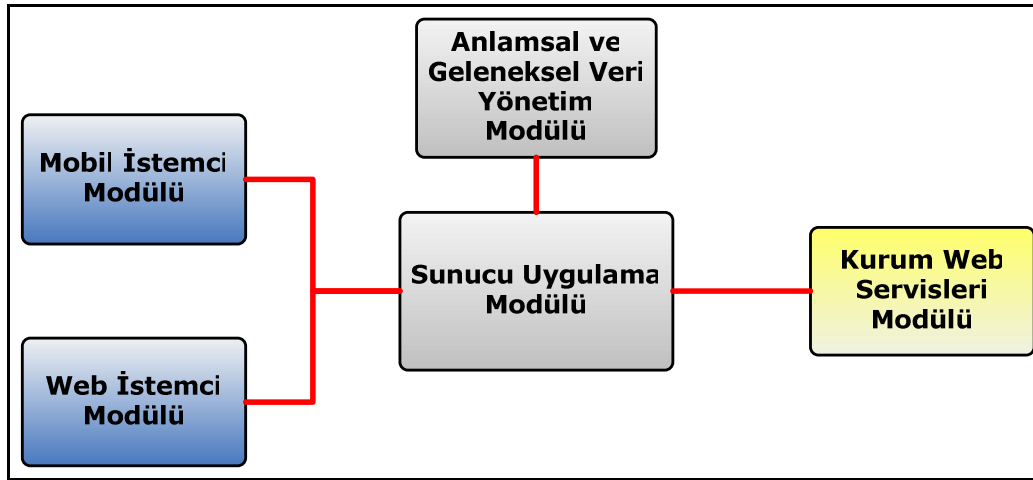
4. ANLAMSAL WEB TEKNİKLERİ KULLANARAK GPS TABANLI BAĞLAM BİLİNÇLİ MOBİL UYGULAMA

Genel olarak istemci-sunucu mimarisine dayanan sisteme MARIS adı verilmiştir. MARIS, “Mobile Augmented Reality Information Services” kısaltmasından oluşan bir takma addır. MARIS platformu beş temel modülden oluşmaktadır.

1. *Mobil istemci modülü*; Mobil cihazlarda çalışan Java Midlet uygulamasıdır. Mobil istemci, kullanıcıya ait dinamik yer bilgisi değerlerini gerçek zamanlı olarak sunucu uygulamasına iletmekte ve sunucunun geri dönüş olarak gönderdiği özel servis bilgilerini kullanıcı arabirimi üzerinde göstermektedir. Kullanıcı ile etkileşimin en yüksek olduğu modüldür.
2. *Web istemci modülü*; Sunucu uygulaması ile gerçek zamanlı olarak haberleşen ve İnternet tarayıcıları üzerinde çalışan bir web uygulamasıdır. Web istemci; yönetici arabirimi ve kullanıcı arabirimi olmak üzere iki farklı bölümden oluşmaktadır. Yönetici arabirimi; kullanıcıların MARIS platformuna giriş bilgilerinin oluşturulması ve yönetilmesi işlevlerinin yerine getirilmesinden sorumludur. Kullanıcı arabirimi ise; kullanıcıya ait profil bilgilerinin oluşturulması ve gerektiğinde değiştirilmesi ile sunulan servislere ait dinamik parametrelerin yönetilmesi görevlerini yürütmektedir.
3. *Sunucu uygulama modülü*; Mobil istemci, web istemci ve kurum web servislerinden gelen istekleri karşılayıp yorumlayarak anlamsal veri modeli ve geleneksel veri yapıları ile haberleşip isteklere uygun cevapları hazırlayan ve bu cevapları geri dönüş olarak istemcilere ileten yapıdır. İçerisinde farklı katmanlarda bir çok uygulama barındırmaktadır. İstemcilerin, gerçekleştirmek istediklere işlere göre uygun sunucu uygulama katmanı ile iletişim kurmaları gerekmektedir..
4. *Anlamsal ve geleneksel veri yönetim modülü*; Kullanıcı profil bilgilerinin geleneksel veri yapıları üzerinde, sunulan servislere ilişkin verilerin ise anlamsal veri modelleri üzerinde depolanıp yönetildiği yapıdır. Veri yapıları ve modelleri veri hacmine göre ölçeklenebilmekte ve dağıtık olarak oluşturulabilmektedir.

5. *Kurum web servisleri modülü*; Sunulan servislere ait verilerin ilgili kurumlardan MARIS platformuna aktarılabilmesi için oluşturulmuş yapıdır. İlgili kurumlar MARIS platformunda yer alan ilgili web servislerini çağırarak kendi taraflarında bulunan verileri MARIS platformuna aktarmaktadırlar.

MARIS platformunu oluşturan bu beş farklı ana modül, sistem içerisinde oluşturulan süreç yönetimine bağlı olarak birbirleri arasında dinamik olarak haberleşmekte ve uygun veri alışverişlerini yapmaktadırlar.



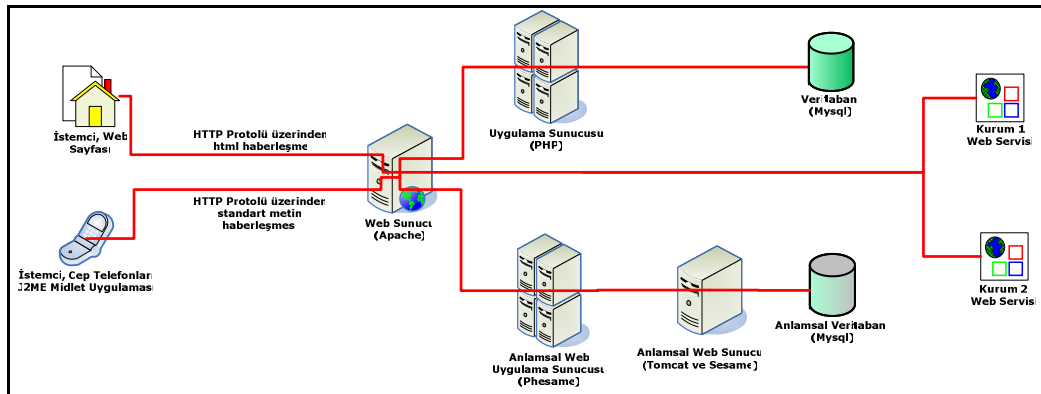
Şekil 4.1. MARIS platformu temel modülleri

Modüller üzerinde işletilen süreç yönetimi aşağıdaki gibidir.

1. Web istemci modülü aracılığı ile kullanıcıya ait hesap bilgileri oluşturulur. Bu hesap bilgileri kullanılarak kullanıcı için uygun profil bilgileri ve servis seçenekleri girilir.
2. Kullanıcı, mobil istemci modülü aracılığı ile dilediği bir an ilgilendiği servis için bağlam bilinçli sonuç talebinde bulunur.
3. Sonuç talebi kullanıcının yer bilgisi ile birlikte sunucu uygulama modülüne ulaşır.

4. Sunucu uygulama modülü anlamsal ve geleneksel veri yönetim modülü ile haberleşerek kullanıcı yer bilgisi, kullanıcı profil bilgileri ve ilgilenilen servise ait verileri değerlendirerek uygun bağlam bilinçli sonucu üretir.
5. Servis sonucu mobil istemci modülüne gönderilir ve kullanıcıyı bilgilendirmek üzere formatlanarak mobil cihaz ekranında gösterilir.
6. Anlamsal ve geleneksel veri yönetim modülünde yer alan servislere ait verilerde herhangi bir değişiklik olacağı zaman kurum web servisleri modülü aracılığı ile ilgili kurumlardan veri güncellemesi yapılır.

4.1. Sistem Mimarisi



Şekil 4.2. MARIS genel mimari

Sistem mimarisi temel bileşenler açısından istemciler ve merkezi sunucudan oluşmaktadır. Bütün bileşenler kendi içerisinde tutarlı olacak şekilde görevini yerine getirmektedir. Her bir sistem bileşeni, platform üzerinde işlenen veri hacmine, kullanıcı sayısına ve işlem miktarına bağlı olarak ölçeklendirilerek yeniden yapılandırılabilir. Her bir sistem bileşeninin ayrı olarak ölçeklendirilebilmesi sayesinde sistem genelinde herhangi bir dar boğaz oluşması engellenebilmekte ve sistemin bütün olarak tutarlı bir şekilde çalışabilmesi sağlanmaktadır. Platformun genelinde oluşabilecek hataların kayıt altına alınması ve daha sonra incelenerek gerekli aksiyonların yerine getirilmesi için uygulama sunucularında işleyen bir kayıt mekanizması bulunmaktadır.

4.1.1. İstemci mimarisi

Bütün istemciler merkezi sistem ile İnternet üzerinden haberleşmektedir. Bunun için istemcilerin uygun İnternet erişimleri olması gerekmektedir.

Mobil İstemci Mimarisi

Mobil cihazlar için geliştirilen uygulamalar cihaz marka ve model bağımlıdır. Bu nedenle çoğu zaman farklı marka ve modeller için farklı geliştirmeler yapmak gerekmektedir. MARIS platformu için mobil cihaz olarak Ericsson markasının P990i modeli kullanılmıştır. P990i modeli mobil cihazlar için özel olarak geliştirilen Symbian işletim sistemi kullanmaktadır.



Resim 4.1. Ericsson marka P990i model mobil cihaz

Cihaz ekranı dokunmatiktir ve standart cep telefonlarına göre daha büyüktür. P990i ayrıca gömülü klavye sistemine sahiptir. Bu özellikleri ile giriş çıkış işlemleri standart cep telefonlarına göre daha kolay olmaktadır.

İnternet bağlantısı için iki farklı metot kullanılabilir. İnternet erişimi için iki farklı metot kullanılabilir.

- GPRS bağlantısı: Hizmet alınan GSM operatörünün müşterilerine İnternet erişimi için sunduğu bir servistir. Kullanım maliyeti vardır.
- Kablosuz ağ bağlantısı: Ulaşılabilir bir kablosuz ağ erişimi mevcut ise kablosuz ağ bağlantısı kurularak İnternet erişimi sağlanabilir. Ek kullanım maliyeti yoktur.

Mobil istemci gerekli yer bilgilerini bluetooth üzerinden haberleştiği harici küresel konumlama sistemi aygıtı aracılığı ile gerçek zamanlı olarak oluşturmaktadır. Küresel konumlama sistemi cihazı RoyalTek firmasının RBT 2210 modelidir.

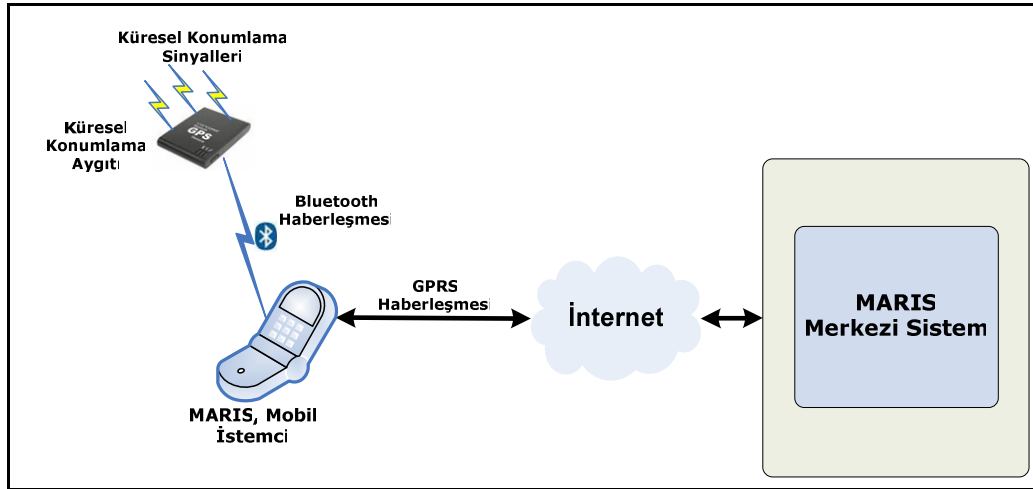


Resim 4.2. RoyalTek marka RBT 2210 model konum belirleme cihazı

Konum belirleme aygıtı, çalışma prensibi nedeni ile kapalı ortamlarda tam performanslı çalışmamaktadır. Böyle durumlarda mobil istemci uygulaması yer bilgilerini boş olarak merkezi sisteme iletmektedir. Merkezi sistem ise bu durumlarda kullanıcı için profilinde saklanan statik konum bilgilerini kullanmaktadır. Mobil istemci uygulaması harici küresel konum belirleme cihazı parametrelerine göre

geliştirildiği için farklı marka mobil cihazlarda bulunan dahili konum belirleme aygıtları üzerinde kullanılamamaktadır. Bunun için mobil istemci yazılımının konum bilgilerini belirleme kısmının yeniden uyarlanması gerekmektedir.

Konum bilgilerini oluşturan mobil istemci yazılımı, bu bilgileri mobil cihazda tanımlanan uygun İnternet bağlantısı aracılığı ile merkezi sisteme iletmektedir. Konum bilgileri merkezi sistem tarafından diğer parametreler ile birlikte değerlendirildikten sonra oluşan sonuç mobil istemci uygulamasına gönderilmektedir. Mobil istemci ilgili servise ait sonucu almakta ve gerekli düzenleme işlemleri sonrasında servis ekranında kullanıcıya sunmaktadır.

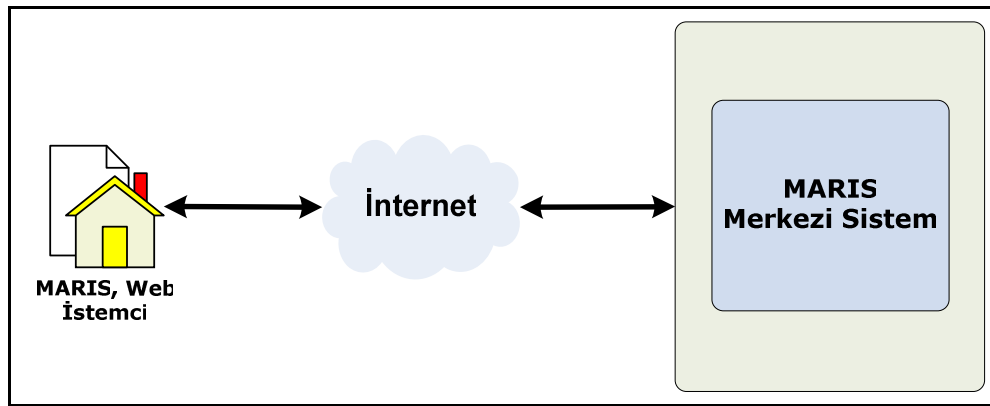


Şekil 4.3. MARIS mobil istemci mimarisi

Mobil istemci ve merkezi sistem haberleşmesi sırasında veri alışverişi HTTP protokolü aracılığı ile yapılmaktadır. İlgili veriyi içeren mesajlar belirlenmiş formatlar doğrultusunda hazırlanmakta ve standart metin olarak oluşturulup HTTP protokolü üzerinde taşınmaktadır. Mobil istemci ve merkezi sistem ilgili mesajları aldıklarında belirlenmiş format doğrultusunda mesajı açabilmekte ve mesaj içerisindeki veriyi uygun şekilde kullanabilmektedirler.

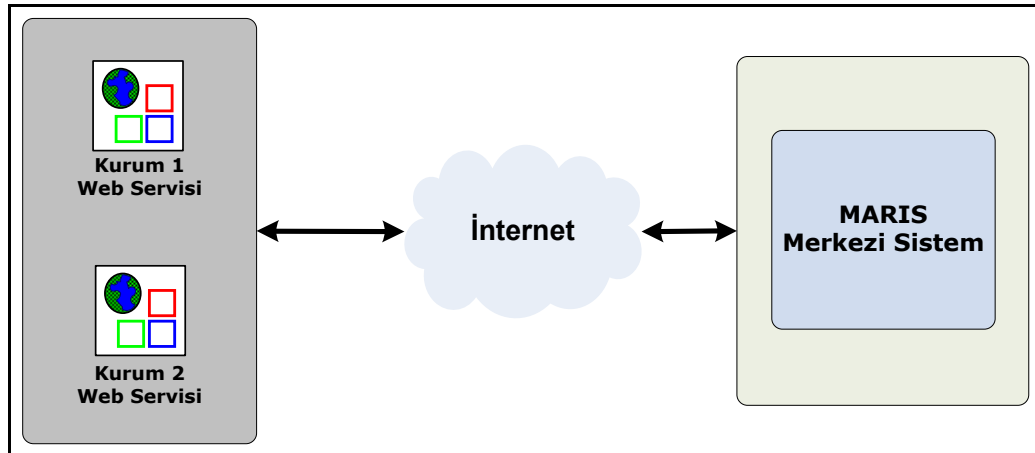
Web İstemci Mimarisi

Yönetici arabirimi ve kullanıcı arabirimi olmak üzere iki kısımdan oluşan web istemci, mevcut İnternet tarayıcıları üzerinde çalışmaktadır. İnternet erişimi bulunan tarayıcı destekli bilgisayarlar aracılığı ile MARIS merkezi sisteme erişilip uygun web arabirimi çağrılmaktadır. Haberleşme protokolü olarak geleneksel web uygulamalarında olduğu gibi HTTP kullanılmaktadır.



Şekil 4.4. MARIS web istemci mimarisi

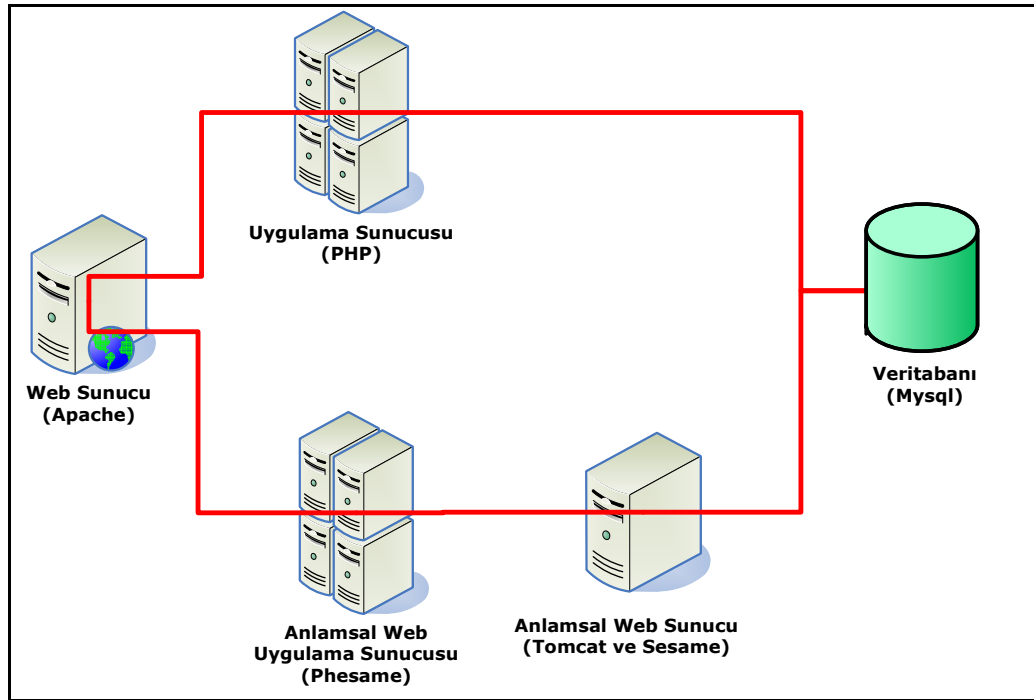
Kurumlar Web Servisi Mimarisi



Şekil 4.5. MARIS kurumlar web servisi mimarisi

4.1.2. Sunucu mimarisi

Sunucu platformu web sunucu, uygulama sunucuları ve veritabanı sunucusundan oluşmaktadır. Bütün sunucular aynı yerel ağda bulunmaktadır. Ancak bu şart değildir. Sunucular farklı yer ve ağlarda bulunabilirler. Yüksek hacimli işlemler için sistemin ölçeklendirilmesi sayesinde dağıtık yapı kurulabilmesi mümkündür.



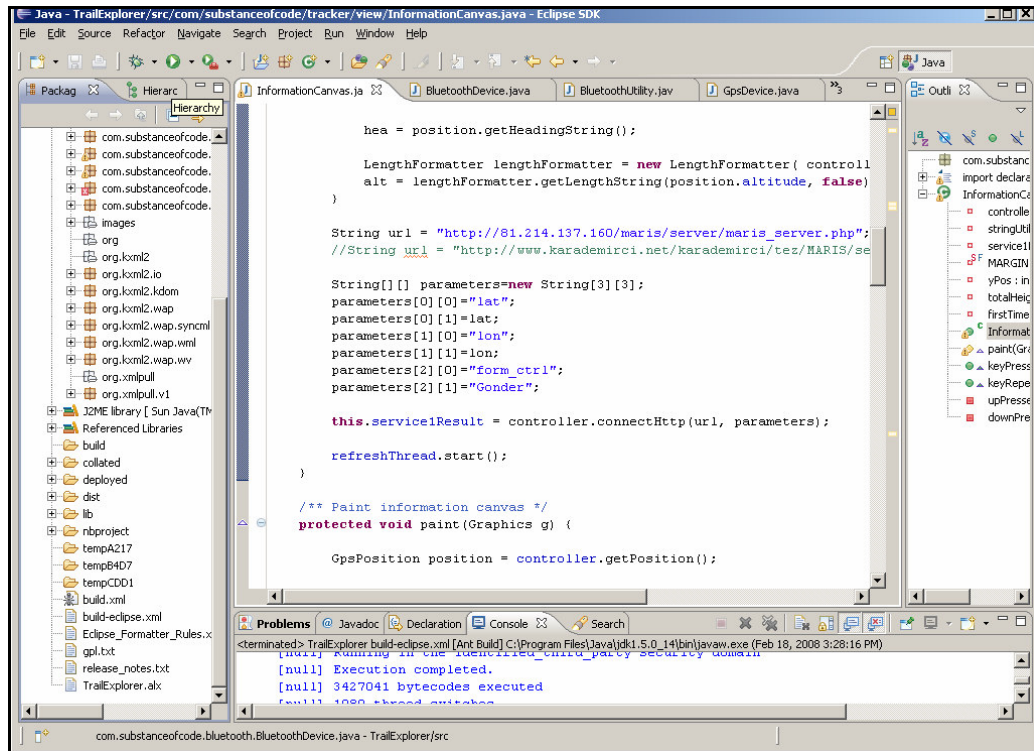
Şekil 4.6. MARIS sunucu mimarisi

Bütün istemcilerden gelen istekler ilk olarak web sunucusu tarafından karşılanmaktadır. Yüksek hacimli işlemler için web sunucu farklı istemciler için özelleştirilebileceği gibi her bir istemci türü içinde birden fazla yapılandırılabilir. Web sunucusu gelen isteğin türüne göre isteği, uygun uygulama sunucusuna iletmektedir. Uygulama sunucusu isteği yorumladıktan sonra cevabı hazırlamak için gerekli verilere erişmek üzere veritabanına ulaşmaktadır. Anlamsal uygulama sunucusu direk veri tabanına erişmeyip anlamsal web sunucusu üzerinden erişim yapmaktadır.

4.2. Mobil İstemci Modülü

Mobil istemci modülü temel olarak, dinamik yer bilgisinin oluşturulması ve bu yer bilgisinin MARIS merkezi sistemine gönderilerek oradan gelecek servis cevabının kullanıcıya sunulması işlemlerini yerine getirmektedir.

Mobil istemci uygulaması Java programlama dilinin mobil cihazlar için var olan ve J2ME olarak adlandırılan versiyonu ile geliştirilmiştir. Geliştirme ortamı olarak Eclipse IDE kullanılmıştır.



Şekil 4.7. Eclipse IDE

Yazılım geliştirme bölümü genel olarak üç farklı kısımdan oluşmaktadır;

- Gerçek zamanlı yer belirleme bölümü
- Merkezi sistem ile veri alışverişinin yapıldığı bölüm
- Kullanıcı arabirimleri

Dinamik Yer Bilgisinin Oluşturulması

Harici küresel yer belirleme cihazı bluetooth bağlantısı üzerinden aranarak bulunmaktadır ve uygun bağlantı gerçekleştirilmektedir.

Bunun için ilk olarak bluetooth servisi hazır hale getirilir.

```
public void initialize() throws BluetoothStateException {
    localDevice = null;
    discoveryAgent = null;
    // Retrieve the local device to get to the Bluetooth
    Manager
    localDevice = LocalDevice.getLocalDevice();
    // Servers set the discoverable mode to GIAC
    localDevice.setDiscoverable(DiscoveryAgent.GIAC);
    // Clients retrieve the discovery agent
    discoveryAgent = localDevice.getDiscoveryAgent();
}
```

Daha sonra bluetooth bağlantısı kurulabilecek cihazların aranması gerçekleştirilir ve bulunan cihazlar kullanıcının seçimine sunulur uygun cihaza bağlantı sağlanır.

```
public void findDevices() {
    try {
        discoveryAgent.startInquiry(DiscoveryAgent.GIAC, this);
        searchComplete = false;
    } catch (BluetoothStateException ex) {
        Logger.getLogger().log(
            "Error in BluetoothUtility.findDevices: " +
            ex.toString(),
            Logger.ERROR);
        ex.printStackTrace();
    }
}
```



```

public void deviceDiscovered(RemoteDevice remoteDevice,
DeviceClass deviceClass) {
    // same device may found several times during single search
    if (devices.indexOf(deviceClass) == -1) {
        String address = remoteDevice.getBluetoothAddress();
        String name = null;
        try {
            name = remoteDevice.getFriendlyName(false);
        } catch (IOException ioe) {}
        if (name == null || name.trim().length() == 0) {
            try {
                name = remoteDevice.getFriendlyName(true);
            } catch (IOException ioe) {}
            if (name == null || name.trim().length() == 0) {
                name = address;
            }
        }
        Logger.getLogger().log(
            "Device found: " + name + " (" + address + ")",
            Logger.INFO);
        BluetoothDevice dev = new BluetoothDevice(address,
name);
        devices.addElement(dev);
    }
}

public synchronized void connect() throws IOException {
    connection = (StreamConnection) Connector.open("btspp://" +
this.getAddress() + ":1",
        Connector.READ);
    reader = new
InputStreamReader(connection.openInputStream());
    thread = new Thread(this);
    thread.start();
}

```

Bağlantı sağlanmasından sonra Küresel Konumlama Sistemi standartlarına göre cihaz ile yapılan haberleşmeden elde edilen veriler uygun şekilde ayrıştırılır ve böylece konum bilgileri oluşturulmuş olur. Bu işlem, istemci yazılımında sürekli tekrarlanır. Böylece konum bilgilerine ne zaman ihtiyaç duyulursa gerçek zamanlı olarak erişilip kullanılmış olur.

```
private synchronized void parseGPGSV(String record) {
    String[] values = StringUtil.split(record, DELIMITER);
    final short cyclePos = Short.parseShort(values[2]);
    if (cyclePos == 1) {
        // New cycle started, copy over last cycles satellites
        and blank.
        copyLastCycleSatellitesAndClear();
    }
    int index = 4;
    while(index+3 < values.length){
        short satelliteNumber = parseShort(values[index++],
GpsSatellite.UNKNOWN);
        short elevation = parseShort(values[index++],
GpsSatellite.UNKNOWN);
        short azimuth = parseShort(values[index++],
GpsSatellite.UNKNOWN);
        short satelliteSnr = parseShort(values[index++],
GpsSatellite.UNKNOWN);
        if(satelliteNumber != GpsSatellite.UNKNOWN){
            final GpsSatellite sat = new
GpsSatellite(satelliteNumber, satelliteSnr, elevation,
azimuth);
            this.tempSatellites.addElement(sat);
        }
    }
}
```

Merkezi Sistem Haberleşmesi

Merkezi sistem ile haberleşme HTTP protokolü üzerinden yapılmaktadır. Merkezi sisteme gönderilecek mesaj hazırlanıp ilgili web sunucuya iletilmekte ve cevap olarak ilgili servis bilgisi beklenmektedir.

```
try {
    long len = 0 ;
    int ch = 0;
    c = (HttpConnection)Connector.open(totalUrl);
    is = c.openInputStream();
    len =c.getLength() ;
    if ( len != -1) {
        // Read exactly Content-Length bytes
        for (int i =0 ; i < len ; i++ )
            if ((ch = is.read()) != -1)
                b.append((char) ch);
    }
    else {
        // Read till the connection is closed.
        while ((ch = is.read()) != -1) {
            len = is.available() ;
            b.append((char)ch);
        }
    }
    this.httpResult = b.toString();
}

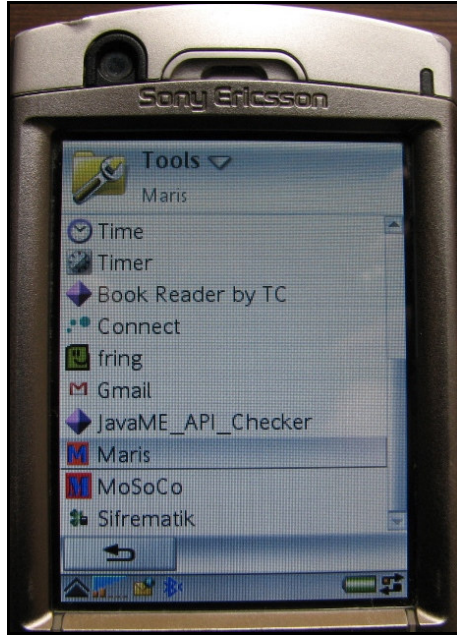
finally {
    is.close();
    c.close();
}
```

Kullanıcı Arabirimleri

Kullanıcı ile etkileşimin yapıldığı bölümlerdir. Genel olarak kullanıcı veri girişi yapmamaktadır. Bunun yerine veriler otomatik olarak oluşturulmaktadır. Bu anlamda yer bilgisi konum belirleme aygıtı aracılığı ile dinamik olarak sağlanmaktadır. Oluşturulan veriler hizmet alınmak istenen servis bilgisi ile birlikte merkezi sisteme gönderilmekte ve merkezi sistemden gelecek cevap beklenmektedir. Merkezi sistemden gelen cevap kullanıcıya sunulmak üzere uygun şekilde formatlanarak ekrana basılmaktadır.

Kullanıcı arabirimleri kullanılan mobil cihaza bağlı olarak bir çok yönden farklılık göstermektedir. Geliştirmelerin yapıldığı mobil cihazda dokunmatik ekran sayesinde kullanıcı gerekli girişleri kolaylıkla yapabilmektedir.

Uygulama mobil cihazda Maris ismi ile yer almaktadır. Uygulamanın başlatılması için Maris aktif edilmelidir.



Resim 4.3. MARIS mobil istemci uygulamasının başlatılması

Maris uygulaması başlatıldığında ilk olarak geçici süreli gösterilen uygulama tanıtıcı arabirim açılmaktadır. Uygulama tanıtıcı arabirim herhangi bir tuşa basılmadığı takdirde altı saniye süre ile beklemektedir. Herhangi bir tuşa basıldığında ise tanıtıcı arabirim kapanmakta ve diğer arabirimlere geçilmektedir.



Resim 4.4. MARIS mobil istemci uygulaması karşılama ekranı

Karşılama arabiriminden sonra durum bilgisinin gösterildiği arabirim açılmaktadır. Durum bilgisi ekranında aynı zamanda kullanıcının işlemler yapacağı menü de yer almaktadır. Uygulama ilk açıldığında durum bilgisi, yer bilgisinin kayıt edilemediğini göstermektedir. Çünkü henüz konum belirleme cihazının uygun bağlantıları sağlanmamıştır.



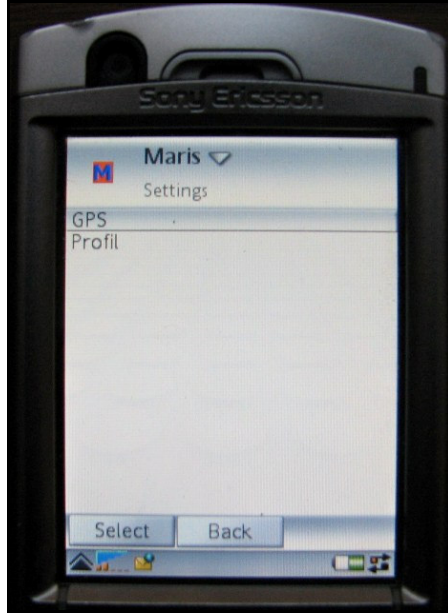
Resim 4.5. MARIS mobil istemci uygulaması durum bilgisi ekranı

Durum bilgisi ekranında yer alan menü seçenekleri sayesinde kullanıcı işlemlerini yapabilmektedir.

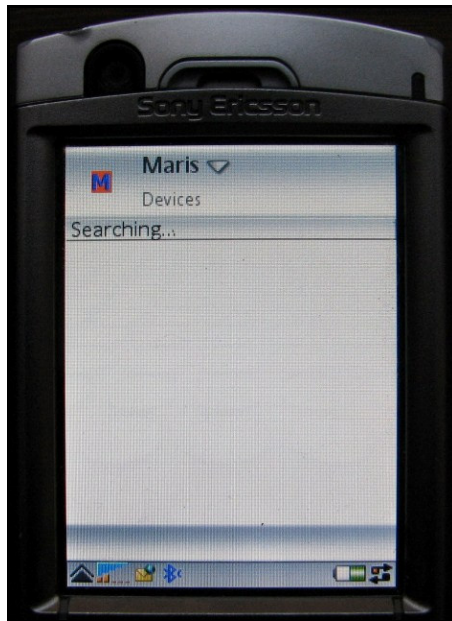


Resim 4.6. MARIS mobil istemci uygulaması menü ekranı

Seeneklerden ilk olarak “Ayarlar” bölümüne girilerek harici konum belirleme cihazının tanıtılması yapılır.

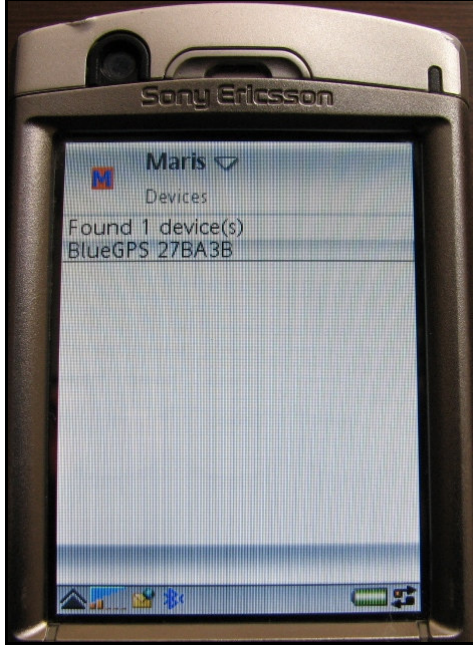


Resim 4.7. MARIS mobil istemci uygulaması ayarlar ekranı



Resim 4.8. MARIS mobil istemci uygulaması konum belirleme cihaz araması

Bluetooth haberleşme protokolü üzerinden bağlanılabilecek aygıtlar arama sonucunda sıralanır ve kullanıcının uygun konum belirleme cihazını seçmesi beklenir.



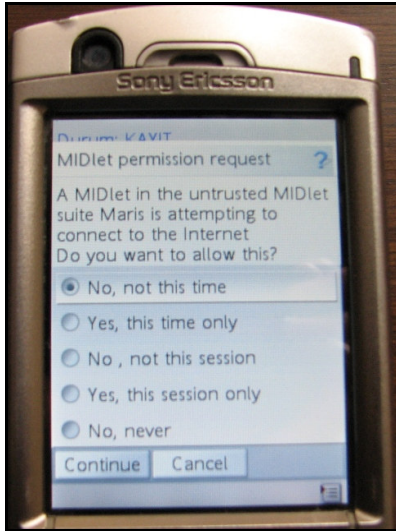
Resim 4.9. MARIS mobil istemci uygulaması bluetooth cihaz arama sonucu

Uygun cihaz seçildikten sonra menüden yer bilgisi verileri için kayıt başlatma seçeneğine girilir. Böylece durum ekranı kayıt yapıldığına dair durum bilgisini değiştirir.



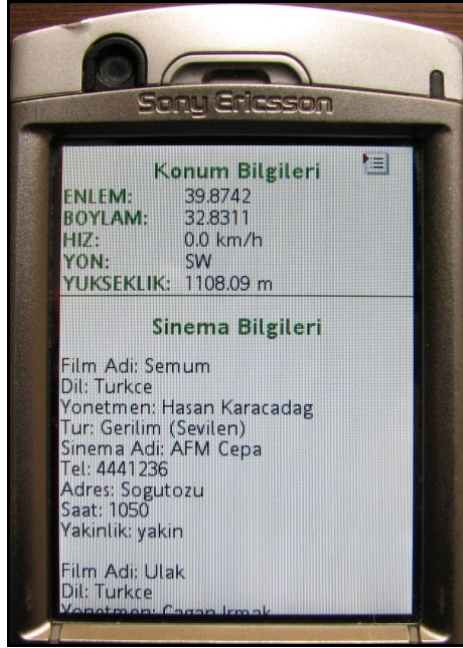
Resim 4.10. MARIS mobil istemci uygulaması durum bilgisi kayıt ekranı

Kayıt işlemi ile birlikte kullanıcı menüden bilgi almak istediği servisi seçer. Servisin İnternet erişimine ihtiyacı olduğu için kullanıcı izni alınır.



Resim 4.11 MARIS mobil istemci uygulaması İnternet erişim izin ekranı

İnternet erişimi sayesinde yer bilgileri MARIS merkezi sistemine gönderilir. Geri dönüş olarak gelen servis ile ilgili sonuç bilgileri de servis bilgi ekranında görüntülenir.



Resim 4.12. MARIS mobil istemci uygulaması servis bilgi ekranı

Kullanıcı istediği zaman menü seçeneklerinden “Yardım” bölümüne girerek uygulama hakkında bilgi ve yardım alabilmektedir.



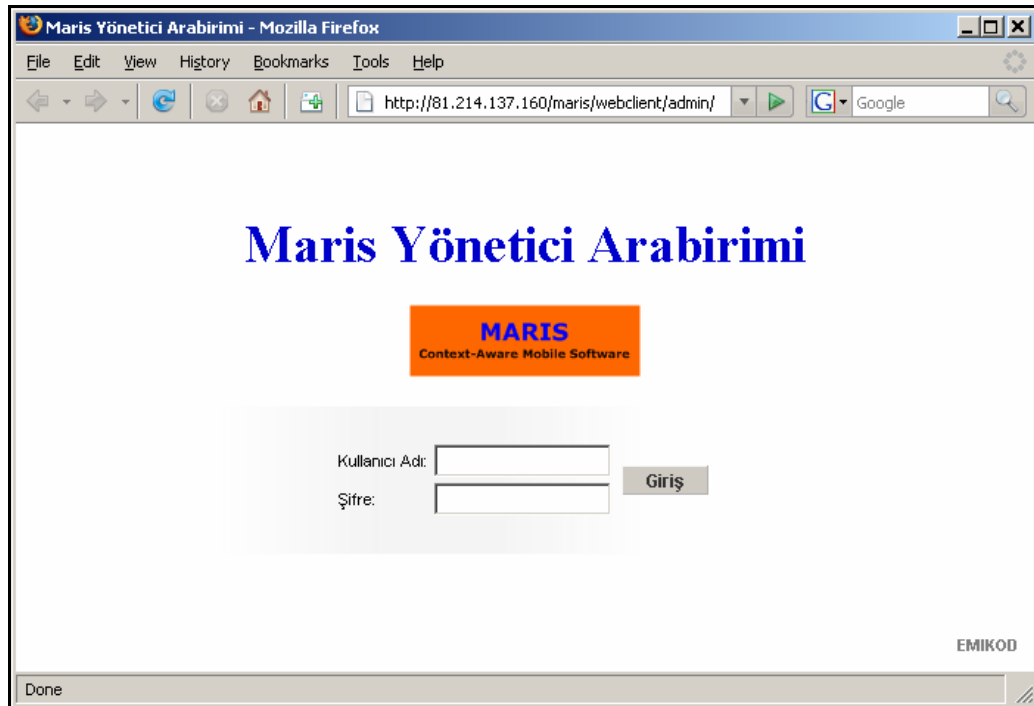
Resim 4.13. MARIS mobil istemci uygulaması yardım ekranı

4.3. Web İstemci Modülü

Web istemci modülü iki farklı arabirimden oluşmaktadır. Web istemci uygulaması sayesinde bilgisayar üzerinde herhangi bir İnternet tarayıcısı kullanılarak MARIS platformuna erişim sağlanıp belli başlı görevlerin yerine getirilmesi mümkündür.

Yönetici Arabirimi

Yönetici arabirimi ile Maris uygulamasını kullanacak kişilerin kullanıcı isimleri ve şifreleri belirlenmektedir. Böylece kullanıcılar kullanıcı arabirimine giriş yapabilmekte ve buradaki işlemleri gerçekleştirebilmektedirler. Yönetici arabirimi birden fazla yöneticiyi desteklemektedir. MARIS platformu üzerinde birden fazla yönetici tanımlanarak yönetici arabirimini kullanmalarına izin verilebilmektedir.

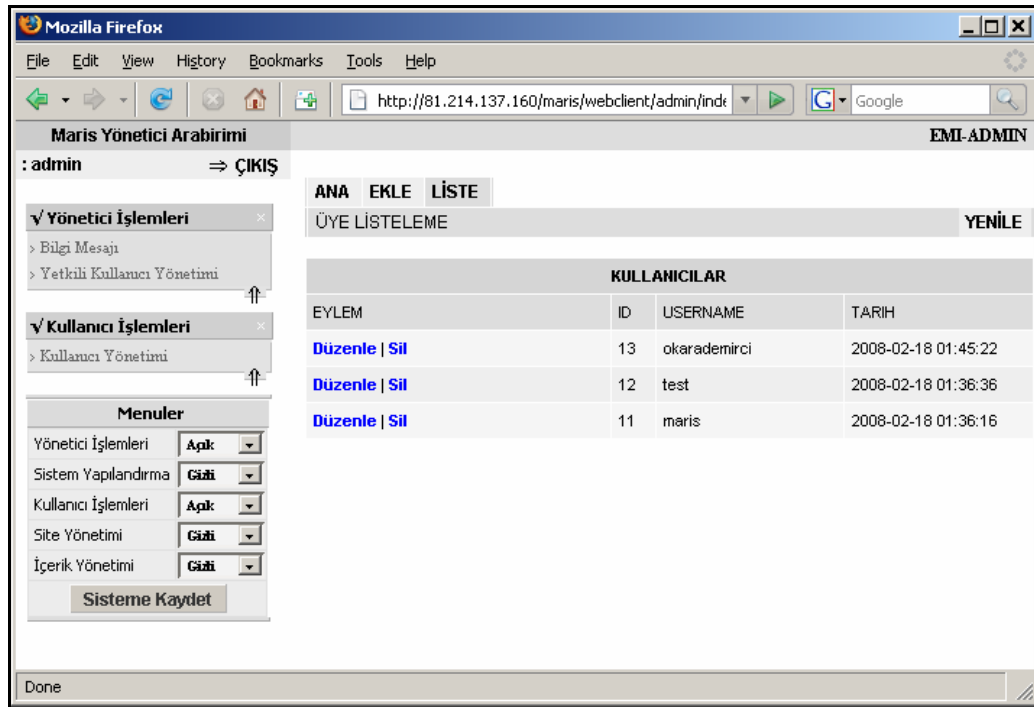


Şekil 4.8.MARIS, web yönetici arabirimi giriş ekranı

Tanımlı yöneticiler giriş yaptıktan sonra yönetici arabirimde bulunan mesajlaşma yapısı ile birbirleri ile haberleşebilmekte ve aralarında bilgi paylaşımı yapabilmektedirler. Bu sayede yöneticiler görev paylaşımı yapabilmekte ve ortak hareket edebilmektedirler.

“Kullanıcı İşlemleri” ana menü başlığı altında bulunan “Kullanıcı Yönetimi” seçeneği ile MARIS platformuna kullanıcı tanımlamaları yapılabilmektedir. Ayrıca var olan kullanıcıların listelenmesi, yeniden düzenlenmesi ve gerekiyorsa silinmesi işlemleri de aynı seçenek altında mevcuttur.

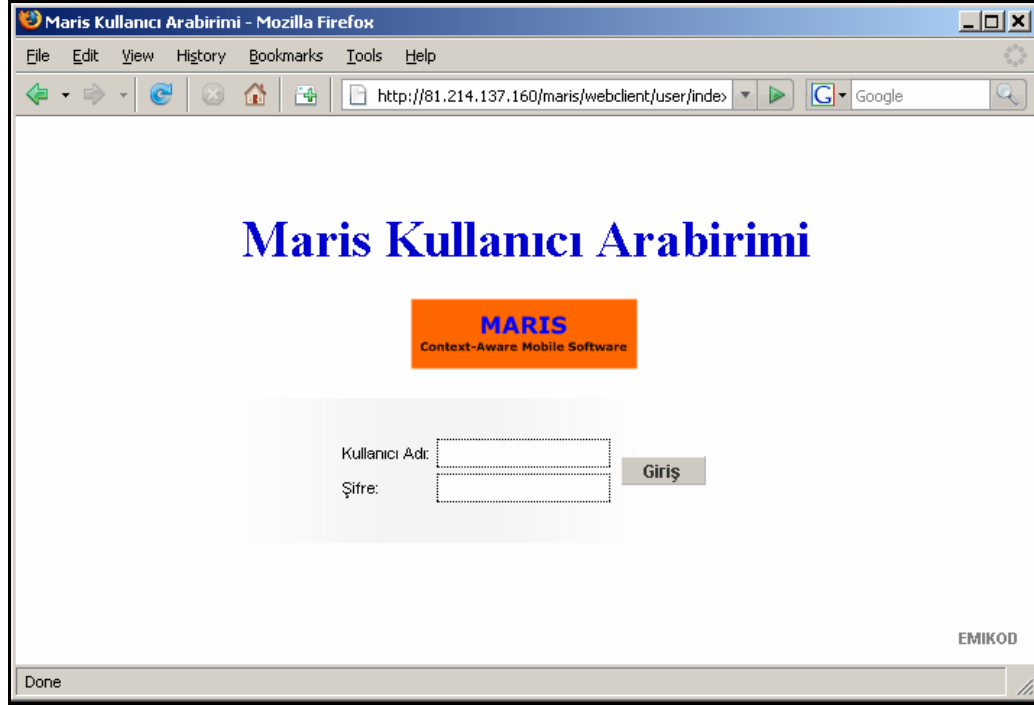
Yönetici kendisine tahsis edilen yönetici şifresini gerek gördüğü taktirde istediği zaman değiştirebilmektedir. Bunun için sol üst kenarda yer alan yönetici ismine giriş yapması yeterlidir.



Şekil 4.9. MARIS, web yönetici arabirimi

Kullanıcı Arabirimi

Yönetici arabiriminde tanımlanan kullanıcı bilgileri ile kullanıcı arabirimine giriş yapılabilir.



Şekil 4.10. MARIS, web kullanıcı arabirimi giriş ekranı

Kullanıcı, kullanıcı arabirimine giriş yaptıktan sonra “Kullanıcı Bilgileri” ve “Sinema Servisi” başlığı altında bulunan kişiye özgü parametreleri değiştirebilmektedir. Kullanıcı bilgileri parametreleri aşağıdaki gibidir.

- Adı ve soyadı,
- Doğum tarihi,
- Şehir,
- Statik enlem bilgisi,
- Statik boylam bilgisi.

Sinema servisi parametreleri aşağıdaki gibidir.

- Sevilen film türü,
- derece yakınlık parametreleri(çok yakın, yakın ve uzak dereceleri)

İlgili parametreler web kullanıcı arabiriminden değiştirildiği anda MARIS platformunda gerekli güncellemeler yapılmaktadır. Böylece mobil istemcide kullanılacak ilgili servisler gerekli bağlam bilinçli sonuç üretimlerinde artık bu yeni parametreleri kullanacaklardır.

Maris Kullanıcı Arabirimi

: maris

Kullanıcı Bilgileri Sinema Servisi

KULLANICI BİLGİLERİ

Kullanıcı Bilgileri

Adı Soyadı: Mobile User

Doğum Tarihi: 23 / 01 / 1979 SEÇ

Şehir: Artvin

Enlem: 39.874

Boylam: 32.8306

Güncelle

Done

Şekil 4.11. MARIS, web kullanıcı arabirimi, kullanıcı bilgileri

4.4. Sunucu Uygulama Modülü

Sunucu uygulaması üç ana kısımdan oluşmaktadır. Tüm kısımlar PHP web programlama dili ile yazılmıştır. PHP, MARIS platformunda web sunucu olarak kullanılan Apache ile entegre bir şekilde çalışmaktadır.

Mobil İstemci Sunucu Kısım

Mobil istemci uygulamasını MARIS merkezi sisteminde ilk karşılayan kısımdır. İlk olarak istemciden gelen parametreleri çözümler. Bu parametreler doğrultusunda geleneksel veritabanında tutulan kullanıcı profiline erişerek gerekli doğrulamaları yapar. Ayrıca bu parametreler doğrultusunda istekte bulunan servis için oluşturulacak bilgileri çıkarmada kullanılacak bağlam parametrelerini oluşturur.

```
//Istemciden gelen yer bilgisi degerlendiriliyor.
$lat_client = $_GET['lat'];
$lon_client = $_GET['lon'];

//mysql veritabanından kullanıcının profil bilgileri alınıyor.
$result=mysql_query("SELECT movie_type, movie_time_interval,
latituted, longituted, distance_param1_interval,
distance_param1_str, distance_param2_interval,
distance_param2_str, distance_param3_interval,
distance_param3_str from md_users where username='maris'",
$mysql_conn);

if(!$result)
{
    $error_line = "Mysql error: " .mysql_errno($mysql_conn)
    .": " .mysql_error($mysql_conn);
    $log_line = "|" .date("c") ."| " . $error_line ."\r\n";
    fputs($fp_log_file, $log_line);
}

list($db_kisi_sinema_turu, $db_zaman_saat_araligi,
$db_yer_bilgisi_enlem, $db_yer_bilgisi_boylam, $db_cok_yakin,
$db_cok_yakin_str, $db_yakin, $db_yakin_str, $db_uzak,
$db_uzak_str)=mysql_fetch_row($result);
```

Oluşturulan bağlam parametreleri ile birlikte servis verilerinin depolandığı anlamsal veritabanı sorgulanır ve servis ile ilgili sonuç oluşturulur. Bu sonuç geri dönüş olarak mobil istemci uygulamasına gönderilir.

Yapılan bütün işlemler kayıt altına alınarak günlük oluşturulan kayıt dosyalarından saklanır. Böylece mobil istemcilerin erişim bilgileri tarihsel olarak oluşturulmuş ve saklanmış olur.

```
$maris_path = dirname(dirname(__FILE__));
$log_path = $maris_path .'/logs/';
$log_file_date = date("d-m-Y");
$log_file_name = "Maris_Server_" . $log_file_date . ".txt";
$fp_log_file = fopen($log_path . $log_file_name, "a") or
die("Cannot write logfile");
$log_line = "-----
-----" . "\r\n";
$log_line .= "|" . date("c") . "| " . "Kayit Basladi." . "\r\n";
fputs($fp_log_file, $log_line);

$log_line = "|" . date("c") . "| " . "Baglam Parametreleri:"
. "\r\n";
$log_line .= "\t" . $baglam_parametreleri . "\r\n";
fputs($fp_log_file, $log_line);

$log_line = "|" . date("c") . "| " . "Istemci icin hazirlanan
cevap:" . "\r\n";
$log_line .= "\t" . $client_str . "\r\n";
fputs($fp_log_file, $log_line);

$log_line = "|" . date("c") . "| " . "Kayit Bitti." . "\r\n";
$log_line .= "-----
-----" . "\r\n";
fputs($fp_log_file, $log_line);
fclose($fp_log_file);
```


Web İstemci Sunucu Kısmı

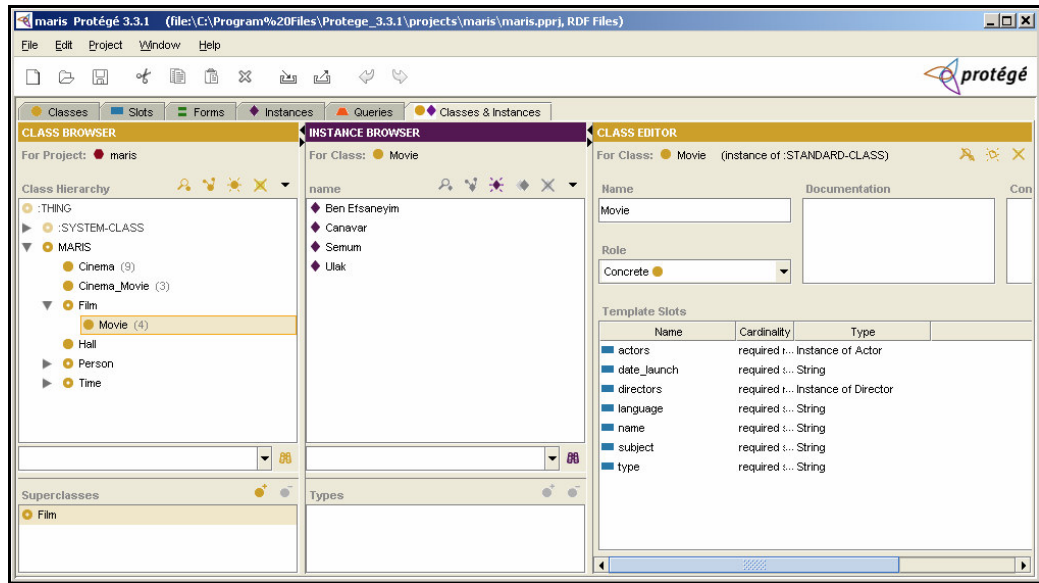
Web istemci uygulamasında yapılan değişiklikler düzenlenerek geleneksel veritabanına yazılır veya ilgili güncellemeler yapılır. Anlamsal veritabanına erişimi bulunmaz, sadece geleneksel veritabanına erişimi vardır.

4.5. Anlamsal ve Geleneksel Veri Yönetim Modülü

Anlamsal veri modelleri ve geleneksel veri yapıları Mysql veritabanı üzerine kurulmuştur. Geleneksel veri yapılarına doğrudan erişim yapılabilirken, anlamsal veri modellerine ise Sesame üzerinden erişilebilmektedir.

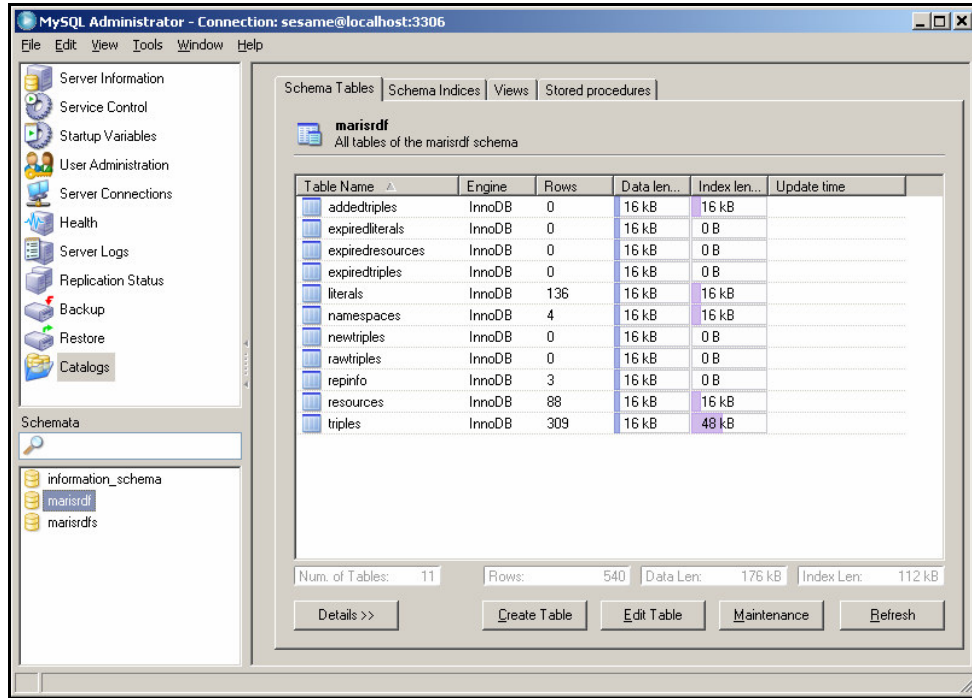
Anlamsal Veri Modelleri

Protege anlamsal web aracı üzerinde oluşturulan RDF ve RDFS veri modelleri Sesame aracılığı ile mysql veritabanına aktarılmıştır.



Şekil 4.12. MARIS Protege RDF veri modeli

RDF ve RDFS veri modellerinin XML tabanlı içerikleri EK-1 ve EK-2’de sunulmuştur. Bu XML tabanlı içerikler Sesame platformunda yer alan web arabirimi aracılığı ile Sesame üzerinden mysql veritabanına aktarılmıştır.



Şekil 4.13. MARIS, RDF veri modeli mysql tabloları

The screenshot shows the MySQL Administrator interface. The left sidebar contains various server management tools. The main window displays the 'marisrdfs' schema, listing 27 tables. The table 'depend' is highlighted, showing 1251 rows and 144 kB of data. The bottom status bar indicates 27 tables, 2,303 rows, 496 kB data length, and 416 kB index length.

Table Name	Engine	Rows	Data len...	Index len...	Update time
addedtriples	InnoDB	0	16 kB	16 kB	
allinferred	InnoDB	0	16 kB	0 B	
allnewtriples	InnoDB	0	16 kB	32 kB	
class	InnoDB	25	16 kB	0 B	
depend	InnoDB	1251	80 kB	144 kB	
direct_subclassof	InnoDB	24	16 kB	16 kB	
direct_subpropertyof	InnoDB	1	16 kB	16 kB	
domain	InnoDB	42	16 kB	16 kB	
expired literals	InnoDB	0	16 kB	0 B	
expiredresources	InnoDB	0	16 kB	0 B	
expiredtriples	InnoDB	0	16 kB	0 B	
groundedtriples	InnoDB	0	16 kB	0 B	
inferred	InnoDB	0	16 kB	0 B	
instanceof	InnoDB	139	16 kB	16 kB	
literals	InnoDB	54	16 kB	16 kB	
namespaces	InnoDB	5	16 kB	16 kB	
newaroundtriples	InnoDB	0	16 kB	0 B	

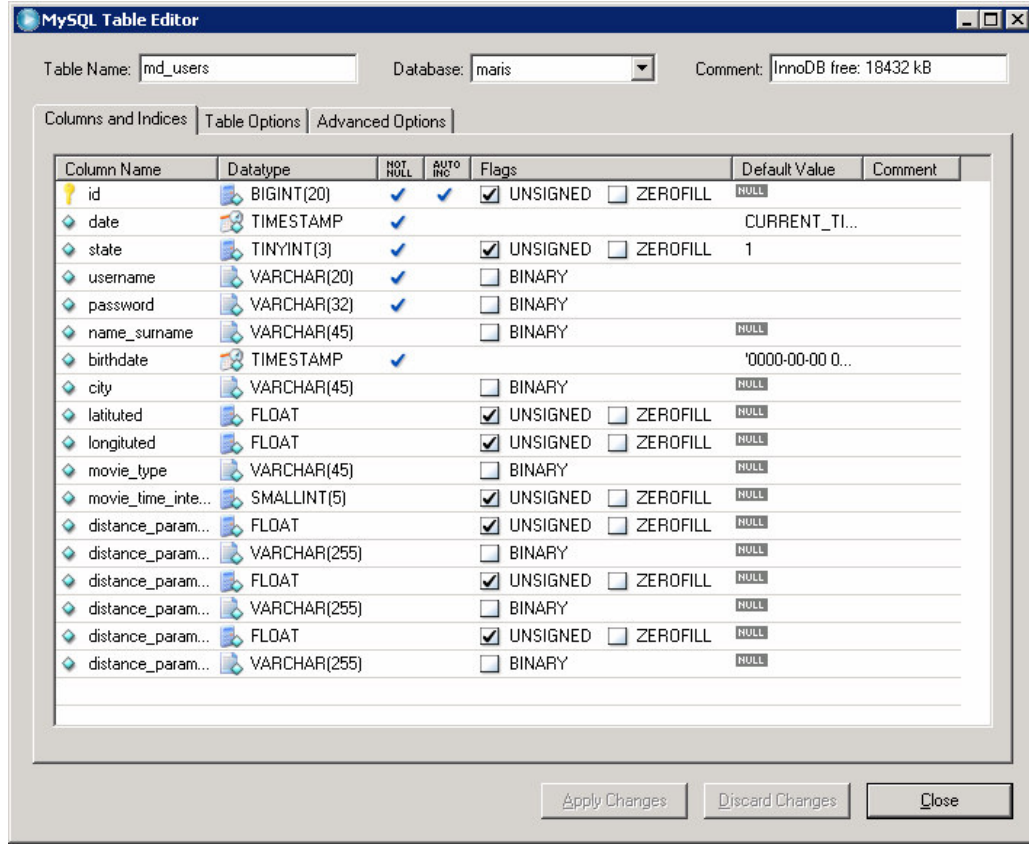
Num. of Tables: 27 Rows: 2,303 Data Len: 496 kB Index Len: 416 kB

Buttons: Details >> Create Table Edit Table Maintenance Refresh

Şekil 4.14.MARIS, RDFS veri modeli mysql tabloları

Geleneksel Veri Yapıları

Geleneksel veri yapıları da mysql veritabanı üzerine kurulmuştur. Web istemci uygulamasının çalışabilmesi için oluşturulmuş bir çok veri yapısı bulunmaktadır. Ancak MARIS platformu için asıl önemli veri yapıları kullanıcı profili için oluşturulmuş yapılardır.



Şekil 4.15. MARIS, kullanıcı profili veri yapıları mysql tablo yapısı

5. SONUÇ VE ÖNERİLER

Bu tez çalışmasında anlamsal web teknikleri kullanılarak bağlam bilinçli mobil bir uygulama geliştirilmiştir. Sistem genel olarak istemci ve sunucu mimarisine dayanmaktadır. İstemciler web istemci ve mobil istemciden oluşmaktadır. Web istemci, kullanıcıların hesap bilgileri, profil bilgileri ve servis tercihlerinin oluşturulmasından sorumludur. Mobil istemci ise kullanıcı yer bilgisinin oluşturulması ve bilgi alınacak ilgili servis seçeneği ile beraber sunuya gönderilmesi görevlerini yerine getirmektedir. Sunucu sistemi ise mobil istemciden gelen yer bilgisi, sunucu sisteminde bulunan kullanıcı profil bilgileri ve anlamsal web teknikleri ile modellenmiş servis verilerini ortak olarak değerlendirip ilgili servise ait bağlam bilinçli sonuçları üretip kullanıcıya sunulmak üzere mobil istemciye göndermektedir.

Geliştirilen uygulama ile daha önce insanlar tarafından oldukça uzun zaman harcanarak elde edilebilen bilgilerin mobil cihaz üzerinde bağlam bilinçli yöntemler kullanılarak otomatik olarak üretilmesi ve kullanıcının hizmetine sunulması sağlanmıştır. Böylece bu uygulama ile kullanıcılara yönelik kişiselleşmiş bilgilere ulaşım hızlanmış, maliyetler düşmüş ve bilgilerin ortak kullanımına yönelik bilgi sınıflandırılması sağlanmıştır.

Geliştirilen bağlam bilinçli mobil uygulamada daha çok uygulama işlevselliğine önem verilmiştir. Bu alanda daha sonra yapılacak çalışmalarda işlevselliğin yanı sıra özellikle ölçeklenebilirlik ve güvenlik konularına da önem verilmelidir. Bu sayede sistemin bütünsel olarak tutarlılığı sağlanmalıdır.

Ayrıca uygulamada kullanılan mobil kullanıcılara ait profil bilgilerinin uyarlanabilir olarak akıllı yöntemlerle güncellenmesi ilerideki çalışmalarda düşünülmelidir. Böylece kullanıcı tercihleri dinamik olarak yönetilebilir ve uygulamanın üreteceği sonuçlar daha kullanışlı olabilir.

Mobil teknolojilerdeki hızlı gelişmeler sonucunda geleneksel bilgisayar uygulamaları yerini gün geçtikçe mobil uygulamalara bırakmaktadır. Mobil uygulamalar her zaman ve her yerde erişilip kullanılabilirler. Bu kapsamda mobil uygulama geliştirmelerine önem verilmeli ve desteklenmelidir.

Anlamsal Web, Web 3.0 olarak adlandırılan ve henüz gelişim aşamasında olan yeni bir yaklaşımdır. Bilgilerin insanlar ve bilgisayarlar tarafından otomatik olarak işlenmesini ve kullanılabilmesini mümkün kılacak bu yaklaşım üzerinde şimdiden çalışmalar başlatılmalı ve desteklenmelidir.

KAYNAKLAR

1. Berners-Lee, T., Cailliau, R., Groff, J.-F. And Pollermann, B., “World-Wide Web: The Information Universe”, *Electronic Networking: Research, Applications and Policy*, 2(1): 52-58 (1992).
2. İnternet : World Wide Web Consortium “Tim Berners-Lee” <http://www.w3.org/People/Berners-Lee/> (2007).
3. İnternet : Patent Storm “United States Patent 5894554, System for managing dynamic web page generation requests by intercepting request at web server and routing to page server thereby releasing web server to process other requests” <http://www.patentstorm.us/patents/5894554.html> (2007).
4. Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N. And Weerawarana, S., “Unraveling the Web Services Web An Introduction to SOAP, WSDL, and UDDI”, *Internet Computing, IEEE*, 6(2): 86-93 (2002).
5. İnternet : World Wide Web Consortium “Extensible Markup Language (XML)” <http://www.w3.org/XML/> (2007).
6. İnternet : World Wide Web Consortium “About W3C, History” <http://www.w3.org/Consortium/history.html> (2007).
7. Muschamp, P., “An introduction to Web Services”, *BT Technology Journal*, 22(1): 9-18 (2004).
8. İnternet : World Wide Web Consortium “Web Services Activity” <http://www.w3.org/2002/ws/> (2007).
9. Casati, F., and Shan M., “Dynamic and adaptive composition of e-services”, *Journal of Information Systems*, 26:143–163 (2001).
10. Berners-Lee, T., Hendler, J. And Lassila, O., “The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities”, *Scientific American*, 284(5): 28–37 (2001).
11. Chae, M. And Kim, J., “What's So Different About the Mobile Internet?”, *Communications of the ACM*, 46(12): 240-247 (2003).
12. İnternet : International Telecommunication Union “Market Information and Statistics” <http://www.itu.int/ITU-D/ict/statistics/ict/index.html> (2007).
13. İnternet : The Nielsen Company “Company Overview” http://www.telephia.com/html/about_telephia.html (2007).

14. Internet : The Nielsen Company “Nielsen Mobile, New Internet and Mobile Measurement Services”
<http://www.telephia.com/html/NielsenMobileRebrand.html> (2007).
15. Wesson, J. L. And van der Walt, D. F., “Implementing Mobile Services: Does the Platform Really Make a Difference?”, *2005 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*, White River, South Africa, 208-216 (2005).
16. Schilit, B.N. And Theimer, M.M., “Disseminating Active Map Information to Mobile Hosts”, *Network, IEEE*, 8(5): 22-32 (1994).
17. Anind, K. D., “Understanding and Using Context”, *Personal and Ubiquitous Computing*, 5(1): 4-7 (2001).
18. Chen, G. And Kotz, D., “A survey of Context-Aware Mobile Computing Research”, *Dartmouth College Computer Science TR2000-381, Hanover, NH, USA*, (2000).
19. D’Roza, T. And Bilchev, G., “An overview of location-based services”, *BT Technology Journal*, 21(1): 20-27 (2003).
20. Iback, P. And Horbank, M., “Highly Available Location-Based Services in Mobile Environments”, *International Service Availability Symposium*, Munich, Germany, (2004).
21. Internet : In-Stat Company “GPS Chip Market Driven by Integration into Mobile Devices”
<http://www.finanznachrichten.de/nachrichten-2007-11/artikel-9385539.asp> (2007).
22. Fabien, L. G. And Norman, M. S., ”Semantic web technology to reconcile privacy and context awareness”, *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 1: 241–260 (2004).
23. Want, R., Hopper, A., Falcão, V. And Gibbons, J., “The Active Badge Location System”, *ACM Trans. Inf. Syst.*, 10(1): 91-102 (1992).
24. Schilit, B., Adams, N. And Want, R., “Context-Aware Computing Applications”, *1st International Workshop on Mobile Computing Systems and Applications*, Santa Cruz, CA, 85–90 (1994).
25. Chou, SC., Hsieh, WT., Gandon, F. L. And Sadeh N. M., “Semantic Web Technologies for Context-Aware Museum Tour Guide Applications”, *19th International Conference on Advanced Information Networking and Applications (AINA’05)*, Tamkang University, Taiwan, 709-714 (2005).

26. Gumus, O., Kardas, G., Dikeneli, O., Erdur, R. C. And Onal A., "SMOP: A Semantic Web and Service Driven Information Gathering Environment for Mobile Platforms", On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE, 4275, **Springer Berlin**, France, 927-940 (2006).
27. Sheshagiri, M., Sadeh, N. M. And Gandon, F., "Using Semantic Web Services for Context-Aware Mobile Applications", **Workshop on Context Awareness, MobiSys 2004**, Boston, (2004).
28. Jeon, JS. And Lee, GJ., "Development of a Semantic Web Based Mobile Local Search System", **16th international conference on World Wide Web**, Banff, Alberta, Canada, 1231-1232 (2007).
29. Cardoso, J., "Semantic Web Services: Theory, Tools and Applications", Klinger, K., Roth, K., Neidig, J., Reed, S., Berger, S. And LeBlanc, J., **IGI Global**, United States of America, United Kingdom, 1-22 (2007).
30. İnternet : Netcraft Company "November 2007 Web Server Survey" http://news.netcraft.com/archives/2007/12/29/december_2007_web_server_survey.html (2008).
31. Berners-Lee, T., Cailliau, R., Groff, J.-F. And Pollermann, B., "World-Wide Web: An Information Infrastructure for High-Energy Physics", **Workshop on Software Engineering, Artificial Intelligence and Expert Systems for High Energy and Nuclear Physics**, La Londe-les- Maures, France, (1992).
32. Challenger J., Iyengar A., Witting K., Ferstat, C. And Reed P., "A Publishing System for Efficiently Creating Dynamic Web Content", **Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies**, Tel Aviv, Israel, 844-853 (2000).
33. İnternet : İnet-tr 2001 "Web Servisleri: İnternet Devriminde İkinci Aşama?" <http://www.research.ibm.com/journal/sj/412/gottschalk.html> (2002).
34. İnternet : IBM "Introduction to Web services architecture" <http://inet-tr.org/inetconf7/program/64.html> (2001).
35. İnternet : SUN "Getting Started on Developing Web Services" http://www.sun.com/bigadmin/content/developer/howtos/developing_webserv.html (2001).
36. Davies, J., Studer, R. And Warren P., "Semantic Web technologies : trends and research in ontology-based systems", Çeviri Editörü/Editörleri, **John Wiley & Sons**, England, 1-34 (2006).

37. Oren, E., Haller, A., Hauswirth, M., Heitmann, B. And Decker, S., “A Flexible Integration Framework for Semantic Web 2.0 Applications”, *Software, IEEE*, 24 (5): 64-71 (2007).
38. Decker, S., Mitra, P. And Menlik, S., “Framework for the Semantic Web: An RDF Tutorial”, *Internet Computing, IEEE*, 4 (6): 68-73 (2000).
39. Antoniou, G. And Harmelen, F. V., “A Semantic Web Primer”, Çeviri Editörü/Editörleri, *The MIT Pres*, London, England, 61-66 (2004).
40. İnternet : Oracle “Anlamsal Teknolojiler Merkezi” http://www.oracle.com/technology/tech/semantic_technologies/index.html (2008).
41. İnternet : Adobe “XMP: Adding intelligence to media” <http://www.adobe.com/products/xmp/index.html> (2008).
42. İnternet : IBM “Integrated Ontology Development Toolkit” <http://www.alphaworks.ibm.com/tech/semanticstk> (2008).
43. İnternet : HP “Semantic Web research labs” <http://www.hpl.hp.com/semweb/> (2008).
44. İnternet : Microsoft “Ms. Dewey” <http://www.msdevey.com/> (2008).
45. İnternet : Hakia “hakia search engine” <http://www.hakia.com/> (2008).
46. İnternet : University of Maryland, Baltimore County “Swoogle, Semantic Web Search Engine” <http://swoogle.umbc.edu/> (2008).
47. İnternet : Aduna open source platform “Sesame” <http://www.openrdf.org/about.jsp> (2008).
48. Broekstra, J., Kampman, A. And Harmelen, F. V., “Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema”, The Semantic Web - ISWC 2002: First International Semantic Web Conference, 2342, *Springer Berlin*, Sardinia, Italy, 54 (2002).
49. İnternet : Aduna open source platform “User Guide for Sesame” <http://www.openrdf.org/doc/sesame/users/userguide.html> (2008).
50. İnternet : Stanford University School of Medicine “Protégé” <http://protege.stanford.edu/> (2008).

51. Noy, N. F., Sintek, M., Decker, S., Crubezy, M., Ferguson, R.W. And Musen, M.A., "Creating Semantic Web contents with Protege-2000", *Intelligent Systems, IEEE*, 16 (2): 60-71 (2001).
52. Costa., P. D., "Towards a Services Platform for Context-Aware Applications", Yüksek Lisans Tezi, *University of Twente Telematics*, Enschede, The Netherlands, 10-35 (2003).
53. Pascoe, J., "Adding generic contextual capabilities to wearable computers", *Second International Symposium on Wearable Computers (ISWC'98)*, Pittsburgh, 92-99 (1998).
54. Salmenjoki, K. And Welzer, T., "Using Web Services and Semantic Web for Producing Intelligent Context-Aware Services", Knowledge-Based Intelligent Information and Engineering Systems, 3214, *Springer Berlin*, New Zealand, 1032-1038 (2004).
55. İnternet : The Nielsen Company "Mobile İnternet Population jumps to 34.6 Million, with email, wheather and sports websites securing the highest reach, according to Telephia" <http://www.telephia.com/documents/İnternetandDeviceReleaseJune2006v68.14.06FINAL.pdf> (2007).
56. İnternet : Openwave "Mobile Browser", http://www.openwave.com/us/products/client_products/mobile_browser/ (2007).
57. İnternet : Motorola "Motorola Mobile İnternet Browser" http://www.motorola.com/mot/doc/1/1049_MotDoc.pdf (2007).
58. İnternet : Nokia "Nokia Web Browser" <http://www.nokia.com/browser> (2007).
59. İnternet : WAPForum "WML, Wireless Markup Language" http://www.wapforum.org/what/WAPWhite_Paper1.pdf (2007).
60. İnternet : Sun Microsystems "Mobil Java ya da J2ME" http://www.java.com/tr/download/faq/whatis_j2me.xml (2007).
61. İnternet : Microsoft ".NET Mobile" <http://www.microsoft.com/windowsmobile/developers/default.aspx> (2007).

* Basılı olarak bulunamayan ve kamu kurum ve kuruluş İnternet sayfaları aracılığı ile erişilemeyen bazı İnternet sayfaları kaynak olarak gösterilmiştir.

EKLER

EK-1 Maris sinema veri modeli RDF şema

```

<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE rdf:RDF [
    <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-
syntax-ns#'>
    <!ENTITY a 'http://protege.stanford.edu/system#'>
    <!ENTITY maris 'http://maris.net/maris#'>
    <!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-
schema#'>
]>
<rdf:RDF xmlns:rdf="&rdf;"
    xmlns:a="&a;"
    xmlns:rdfs="&rdfs;"
    xmlns:maris="&maris;">
<rdfs:Class rdf:about="&maris;Actor"
    rdfs:label="Actor">
    <rdfs:subClassOf rdf:resource="&maris;Person"/>
</rdfs:Class>
<rdfs:Class rdf:about="&maris;Cinema"
    rdfs:label="Cinema">
    <rdfs:subClassOf rdf:resource="&maris;MARIS"/>
</rdfs:Class>
<rdfs:Class rdf:about="&maris;Cinema_Movie"
    rdfs:label="Cinema_Movie">
    <rdfs:subClassOf rdf:resource="&maris;MARIS"/>
</rdfs:Class>
<rdfs:Class rdf:about="&maris;Director"
    rdfs:label="Director">
    <rdfs:subClassOf rdf:resource="&maris;Person"/>
</rdfs:Class>
<rdfs:Class rdf:about="&maris;Film"
    a:role="abstract"
    rdfs:label="Film">
    <rdfs:subClassOf rdf:resource="&maris;MARIS"/>
</rdfs:Class>
<rdfs:Class rdf:about="&maris;Hall"
    rdfs:label="Hall">
    <rdfs:subClassOf rdf:resource="&maris;MARIS"/>
</rdfs:Class>
<rdfs:Class rdf:about="&maris;MARIS"
    a:role="abstract"
    rdfs:comment="Mobile Augmented Reality Information
Services, Main Class"
    rdfs:label="MARIS">
    <rdfs:subClassOf rdf:resource="&rdfs;Resource"/>

```

EK-1 (Devam) Maris sinema veri modeli RDF şema

```

</rdfs:Class>
<rdfs:Class rdf:about="&maris;Movie"
    rdfs:label="Movie">
    <rdfs:subClassOf rdf:resource="&maris;Film"/>
</rdfs:Class>
<rdfs:Class rdf:about="&maris;Person"
    a:role="abstract"
    rdfs:label="Person">
    <rdfs:subClassOf rdf:resource="&maris;MARIS"/>
</rdfs:Class>
<rdfs:Class rdf:about="&maris;Time"
    a:role="abstract"
    rdfs:label="Time">
    <rdfs:subClassOf rdf:resource="&maris;MARIS"/>
</rdfs:Class>
<rdfs:Class rdf:about="&maris;Time_Interval"
    rdfs:label="Time_Interval">
    <rdfs:subClassOf rdf:resource="&maris;Time"/>
</rdfs:Class>
<rdfs:Class rdf:about="&maris;Week_Day"
    rdfs:label="Week_Day">
    <rdfs:subClassOf rdf:resource="&maris;Time"/>
</rdfs:Class>
<rdf:Property rdf:about="&maris;actors"
    a:minCardinality="1"
    rdfs:label="actors">
    <rdfs:range rdf:resource="&maris;Actor"/>
    <rdfs:domain rdf:resource="&maris;Movie"/>
</rdf:Property>
<rdf:Property rdf:about="&maris;address"
    a:maxCardinality="1"
    rdfs:label="address">
    <rdfs:domain rdf:resource="&maris;Cinema"/>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
</rdf:Property>
<rdf:Property rdf:about="&maris;capacity"
    a:maxCardinality="1"
    a:range="integer"
    rdfs:label="capacity">
    <rdfs:domain rdf:resource="&maris;Hall"/>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
</rdf:Property>
<rdf:Property rdf:about="&maris;cinema"
    a:maxCardinality="1"
    a:minCardinality="1"

```

EK-1 (Devam) Maris sinema veri modeli RDF şema

```

        rdfs:label="cinema">
        <rdfs:range rdf:resource="&maris;Cinema"/>
    </rdf:Property>
    <rdf:Property rdf:about="&maris;cinemas"
        a:maxCardinality="1"
        a:minCardinality="1"
        rdfs:label="cinemas">
        <rdfs:range rdf:resource="&maris;Cinema"/>
        <rdfs:domain rdf:resource="&maris;Cinema_Movie"/>
    </rdf:Property>
    <rdf:Property rdf:about="&maris;city"
        a:maxCardinality="1"
        a:minCardinality="1"
        rdfs:label="city">
        <rdfs:domain rdf:resource="&maris;Cinema"/>
        <rdfs:range rdf:resource="&rdfs;Literal"/>
    </rdf:Property>
    <rdf:Property rdf:about="&maris;date_launch"
        a:maxCardinality="1"
        a:minCardinality="1"
        rdfs:label="date_launch">
        <rdfs:domain rdf:resource="&maris;Movie"/>
        <rdfs:range rdf:resource="&rdfs;Literal"/>
    </rdf:Property>
    <rdf:Property rdf:about="&maris;directors"
        a:minCardinality="1"
        rdfs:label="directors">
        <rdfs:range rdf:resource="&maris;Director"/>
        <rdfs:domain rdf:resource="&maris;Movie"/>
    </rdf:Property>
    <rdf:Property rdf:about="&maris;hall"
        rdfs:label="hall">
        <rdfs:domain rdf:resource="&maris;Cinema"/>
        <rdfs:range rdf:resource="&maris;Hall"/>
    </rdf:Property>
    <rdf:Property rdf:about="&maris;halls"
        a:maxCardinality="1"
        rdfs:label="halls">
        <rdfs:domain rdf:resource="&maris;Cinema_Movie"/>
        <rdfs:range rdf:resource="&maris;Hall"/>
    </rdf:Property>
    <rdf:Property rdf:about="&maris;intervals"
        a:maxCardinality="1"
        a:minCardinality="1"
        rdfs:label="intervals">

```

EK-1 (Devam) Maris sinema veri modeli RDF şema

```

        <rdfs:domain rdf:resource="&maris;Time_Interval"/>
        <rdfs:range rdf:resource="&rdfs;Literal"/>
    </rdf:Property>
    <rdf:Property rdf:about="&maris;language"
        a:maxCardinality="1"
        a:minCardinality="1"
        rdfs:label="language">
        <rdfs:domain rdf:resource="&maris;Movie"/>
        <rdfs:range rdf:resource="&rdfs;Literal"/>
    </rdf:Property>
    <rdf:Property rdf:about="&maris;latitude"
        a:maxCardinality="1"
        a:minCardinality="1"
        a:range="float"
        rdfs:label="latitude">
        <rdfs:domain rdf:resource="&maris;Cinema"/>
        <rdfs:range rdf:resource="&rdfs;Literal"/>
    </rdf:Property>
    <rdf:Property rdf:about="&maris;longitude"
        a:maxCardinality="1"
        a:minCardinality="1"
        a:range="float"
        rdfs:label="longitude">
        <rdfs:domain rdf:resource="&maris;Cinema"/>
        <rdfs:range rdf:resource="&rdfs;Literal"/>
    </rdf:Property>
    <rdf:Property rdf:about="&maris;movies"
        a:maxCardinality="1"
        a:minCardinality="1"
        rdfs:label="movies">
        <rdfs:domain rdf:resource="&maris;Cinema_Movie"/>
        <rdfs:range rdf:resource="&maris;Movie"/>
    </rdf:Property>
    <rdf:Property rdf:about="&maris;name"
        a:maxCardinality="1"
        a:minCardinality="1"
        rdfs:label="name">
        <rdfs:domain rdf:resource="&maris;Actor"/>
        <rdfs:domain rdf:resource="&maris;Cinema"/>
        <rdfs:domain rdf:resource="&maris;Director"/>
        <rdfs:domain rdf:resource="&maris;Hall"/>
        <rdfs:domain rdf:resource="&maris;Movie"/>
        <rdfs:range rdf:resource="&rdfs;Literal"/>
    </rdf:Property>
    <rdf:Property rdf:about="&maris;other_info"

```


EK-1 (Devam) Maris sinema veri modeli RDF şema

```

        a:maxCardinality="1"
        rdfs:label="other_info">
<rdfs:domain rdf:resource="&maris;Actor"/>
<rdfs:domain rdf:resource="&maris;Director"/>
<rdfs:range rdf:resource="&rdfs;Literal"/>
</rdf:Property>
<rdf:Property rdf:about="&maris;phone_number"
        a:maxCardinality="1"
        rdfs:label="phone_number">
<rdfs:domain rdf:resource="&maris;Cinema"/>
<rdfs:range rdf:resource="&rdfs;Literal"/>
</rdf:Property>
<rdf:Property rdf:about="&maris;seance_days"
        a:allowedValues="Wednesday"
        a:minCardinality="1"
        a:range="symbol"
        rdfs:label="seance_days">
<rdfs:domain rdf:resource="&maris;Cinema_Movie"/>
<rdfs:range rdf:resource="&rdfs;Literal"/>
<a:allowedValues>Friday</a:allowedValues>
<a:allowedValues>Monday</a:allowedValues>
<a:allowedValues>Saturday</a:allowedValues>
<a:allowedValues>Sunday</a:allowedValues>
<a:allowedValues>Thursday</a:allowedValues>
<a:allowedValues>Tuesday</a:allowedValues>
</rdf:Property>
<rdf:Property rdf:about="&maris;seance_times"
        a:minCardinality="1"
        a:range="integer"
        rdfs:label="seance_times">
<rdfs:domain rdf:resource="&maris;Cinema_Movie"/>
<rdfs:range rdf:resource="&rdfs;Literal"/>
</rdf:Property>
<rdf:Property rdf:about="&maris;subject"
        a:maxCardinality="1"
        a:minCardinality="1"
        rdfs:label="subject">
<rdfs:domain rdf:resource="&maris;Movie"/>
<rdfs:range rdf:resource="&rdfs;Literal"/>
</rdf:Property>
<rdf:Property rdf:about="&maris;type"
        a:maxCardinality="1"
        a:minCardinality="1"
        rdfs:label="type">
<rdfs:domain rdf:resource="&maris;Movie"/>

```

EK-1 (Devam) Maris sinema veri modeli RDF şema

```

        <rdfs:range rdf:resource="&rdfs;Literal"/>
    </rdf:Property>
    <rdf:Property rdf:about="&maris;week_days"
        a:allowedValues="Sali"
        a:maxCardinality="1"
        a:minCardinality="1"
        a:range="symbol"
        rdfs:label="week_days">
        <rdfs:domain rdf:resource="&maris;Week_Day"/>
        <rdfs:range rdf:resource="&rdfs;Literal"/>
        <a:allowedValues>Carsamba</a:allowedValues>
        <a:allowedValues>Cuma</a:allowedValues>
        <a:allowedValues>Cumartesi</a:allowedValues>
        <a:allowedValues>Pazar</a:allowedValues>
        <a:allowedValues>Pazartesi</a:allowedValues>
        <a:allowedValues>Persembes</a:allowedValues>
    </rdf:Property>
</rdf:RDF>

```

EK-2 Maris sinema veri modeli RDF

```

<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE rdf:RDF [
    <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-
syntax-ns# '>
    <!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-
schema# '>
    <!ENTITY maris 'http://maris.net/maris# '>
]>
<rdf:RDF xmlns:rdf="&rdf;"
    xmlns:maris="&maris;"
    xmlns:rdfs="&rdfs;">
<maris:Movie rdf:about="&maris;maris_Instance_10"
    maris:date_launch="15/02/2008"
    maris:language="Ingilizce"
    maris:name="Canavar"
    maris:type="Macera"
    rdfs:label="Canavar">
    <maris:subject>New Yorktaki bir barda kulaklari sagir
eden bir gurultu duyulur, bardaki kargasa sirasinda
davetliler merdivenden asagiya kacmaya calisirken New
York caddelerini alev alev yanan yikinti ve enkazlar
kaplar.</maris:subject>
    <maris:directors
rdf:resource="&maris;maris_Instance_11"/>
    <maris:actors
rdf:resource="&maris;maris_Instance_12"/>
    <maris:actors
rdf:resource="&maris;maris_Instance_13"/>
    <maris:actors
rdf:resource="&maris;maris_Instance_14"/>
    <maris:actors
rdf:resource="&maris;maris_Instance_15"/>
</maris:Movie>
<maris:Time_Interval
rdf:about="&maris;maris_Instance_100"
    maris:intervals="16:30"
    rdfs:label="16:30"/>
<maris:Time_Interval
rdf:about="&maris;maris_Instance_101"
    maris:intervals="19:20"
    rdfs:label="19:20"/>
<maris:Time_Interval
rdf:about="&maris;maris_Instance_104"
    maris:intervals="22:10"
    rdfs:label="22:10"/>

```

EK-2 (Devam) Maris sinema veri modeli RDF

```

<maris:Director rdf:about="&maris;maris_Instance_11"
    maris:name="Matt Reeves"
    rdfs:label="Matt Reeves"/>
<maris:Cinema_Movie rdf:about="&maris;maris_Instance_117"
    maris:seance_days="Wednesday"
    rdfs:label="Semum-AFM Cepa">
    <maris:cinemas
rdf:resource="&maris;maris_Instance_46"/>
    <maris:movies
rdf:resource="&maris;maris_Instance_97"/>
    <maris:seance_days>Friday</maris:seance_days>
    <maris:seance_days>Monday</maris:seance_days>
    <maris:seance_days>Saturday</maris:seance_days>
    <maris:seance_days>Sunday</maris:seance_days>
    <maris:seance_days>Thursday</maris:seance_days>
    <maris:seance_days>Tuesday</maris:seance_days>
    <maris:seance_times>1050</maris:seance_times>
    <maris:seance_times>1335</maris:seance_times>
    <maris:seance_times>1615</maris:seance_times>
    <maris:seance_times>1845</maris:seance_times>
    <maris:seance_times>2120</maris:seance_times>
    <maris:seance_times>2345</maris:seance_times>
</maris:Cinema_Movie>
<maris:Week_Day rdf:about="&maris;maris_Instance_118"
    maris:week_days="Pazartesi"
    rdfs:label="Pazartesi"/>
<maris:Week_Day rdf:about="&maris;maris_Instance_119"
    maris:week_days="Sali"
    rdfs:label="Sali"/>
<maris:Actor rdf:about="&maris;maris_Instance_12"
    maris:name="T.J. Miller"
    rdfs:label="T.J. Miller"/>
<maris:Week_Day rdf:about="&maris;maris_Instance_120"
    maris:week_days="Carsamba"
    rdfs:label="Carsamba"/>
<maris:Week_Day rdf:about="&maris;maris_Instance_121"
    maris:week_days="Persembes"
    rdfs:label="Persembes"/>
<maris:Week_Day rdf:about="&maris;maris_Instance_122"
    maris:week_days="Cuma"
    rdfs:label="Cuma"/>
<maris:Week_Day rdf:about="&maris;maris_Instance_123"
    maris:week_days="Cumartesi"
    rdfs:label="Cumartesi"/>
<maris:Week_Day rdf:about="&maris;maris_Instance_124"

```

EK-2 (Devam) Maris sinema veri modeli RDF

```

        maris:week_days="Pazar"
        rdfs:label="Pazar"/>
<maris:Actor rdf:about="&maris;maris_Instance_13"
    maris:name="Michael Stahl-David"
    rdfs:label="Michael Stahl-David"/>
<maris:Actor rdf:about="&maris;maris_Instance_14"
    maris:name="Mike Vogel"
    rdfs:label="Mike Vogel"/>
<maris:Actor rdf:about="&maris;maris_Instance_15"
    maris:name="Lizzy Caplan"
    rdfs:label="Lizzy Caplan"/>
<maris:Cinema_Movie rdf:about="&maris;maris_Instance_16"
    maris:seance_times="2230"
    rdfs:label="Canavar-Armada">
    <maris:movies
rdf:resource="&maris;maris_Instance_10"/>
    <maris:cinemas
rdf:resource="&maris;maris_Instance_51"/>
        <maris:seance_days>Friday</maris:seance_days>
        <maris:seance_days>Monday</maris:seance_days>
        <maris:seance_days>Sunday</maris:seance_days>
        <maris:seance_days>Thursday</maris:seance_days>
        <maris:seance_days>Tuesday</maris:seance_days>
        <maris:seance_days>Wednesday</maris:seance_days>
        <maris:seance_times>1145</maris:seance_times>
        <maris:seance_times>1345</maris:seance_times>
        <maris:seance_times>1545</maris:seance_times>
        <maris:seance_times>1745</maris:seance_times>
        <maris:seance_times>1945</maris:seance_times>
        <maris:seance_times>2145</maris:seance_times>
    </maris:Cinema_Movie>
<maris:Movie rdf:about="&maris;maris_Instance_3"
    maris:date_launch="25/01/2008"
    maris:language="Ingilizce"
    maris:name="Ben Efsaneyim"
    maris:type="Bilim kurgu"
    rdfs:label="Ben Efsaneyim">
    <maris:subject>Robert Neville cok basarili bir bilim
adami olsa da; o bile, durdurulamayan, tedavi edilemeyen,
insan yapimi o korkunc virusu kontrol altina
alamamistir.</maris:subject>
    <maris:directors
rdf:resource="&maris;maris_Instance_4"/>
    <maris:actors
rdf:resource="&maris;maris_Instance_5"/>

```

EK-2 (Devam) Maris sinema veri modeli RDF

```

    <maris:actors
rdf:resource="&maris;maris_Instance_6"/>
    <maris:actors
rdf:resource="&maris;maris_Instance_7"/>
    <maris:actors
rdf:resource="&maris;maris_Instance_8"/>
</maris:Movie>
<maris:Director rdf:about="&maris;maris_Instance_4"
    maris:name="Francis Lawrence"
    rdfs:label="Francis Lawrence"/>
<maris:Cinema rdf:about="&maris;maris_Instance_46"
    maris:address="Sogutozu"
    maris:city="Ankara"
    maris:latitude="39.6421"
    maris:longitude="32.7112"
    maris:name="AFM Ceba"
    maris:phone_number="4441236"
    rdfs:label="AFM Ceba"/>
<maris:Cinema rdf:about="&maris;maris_Instance_47"
    maris:address="Migros Alisveris Merkezi No:2/242
Akkopru"
    maris:city="Ankara"
    maris:latitude="39.1278"
    maris:longitude="32.1654"
    maris:name="AFM IMAX"
    maris:phone_number="4441236"
    rdfs:label="AFM IMAX"/>
<maris:Cinema rdf:about="&maris;maris_Instance_48"
    maris:address="Bahcelievler"
    maris:city="Anakra"
    maris:latitude="40.1348"
    maris:longitude="33.2132"
    maris:name="AFM ON"
    maris:phone_number="03122153400"
    rdfs:label="AFM ON"/>
<maris:Cinema rdf:about="&maris;maris_Instance_49"
    maris:address="Kizilirmak sok. No:14 Kocatepe"
    maris:city="Ankara"
    maris:latitude="39.7653"
    maris:longitude="32.5352"
    maris:name="Ankapol"
    maris:phone_number="03124193959"
    rdfs:label="Ankapol"/>
<maris:Actor rdf:about="&maris;maris_Instance_5"
    maris:name="Will Smith"

```

EK-2 (Devam) Maris sinema veri modeli RDF

```

        rdfs:label="Will Smith"/>
<maris:Cinema rdf:about="&maris;maris_Instance_50"
    maris:address="Tunali Hilmi Cad. 85/31 Kavaklidere"
    maris:city="Ankara"
    maris:latitude="38.78645"
    maris:longitude="31.8121"
    maris:name="Ankara Sinema Kulturu Dernegi"
    maris:phone_number="03124672002"
    rdfs:label="Ankara Sinema Kulturu Dernegi"/>
<maris:Cinema rdf:about="&maris;maris_Instance_51"
    maris:address="Armada Alisveris Merkezi Eskisehir
yolu No:6 Sogutozu"
    maris:city="Ankara"
    maris:latitude="38.3126"
    maris:longitude="31.2324"
    maris:name="Armada"
    maris:phone_number="03122191600"
    rdfs:label="Armada"/>
<maris:Cinema rdf:about="&maris;maris_Instance_52"
    maris:address="Ataturk Bulvari No: 151 Bakanlik"
    maris:city="Ankara"
    maris:latitude="40.3451"
    maris:longitude="33.2451"
    maris:name="Bati"
    maris:phone_number="03124183028"
    rdfs:label="Bati"/>
<maris:Cinema rdf:about="&maris;maris_Instance_53"
    maris:address="Konur Sok. No:33 Mesrutiyet Kizilay"
    maris:city="Ankara"
    maris:latitude="39.874"
    maris:longitude="32.8306"
    maris:name="Megapol"
    maris:phone_number="03124194492"
    rdfs:label="Megapol"/>
<maris:Cinema rdf:about="&maris;maris_Instance_54"
    maris:address="Selanik Cad. No:76 Kizilay"
    maris:city="Ankara"
    maris:latitude="39.6579"
    maris:longitude="32.5674"
    maris:name="Metropol"
    maris:phone_number="03124257478"
    rdfs:label="Metropol"/>
<maris:Actor rdf:about="&maris;maris_Instance_6"
    maris:name="Alice Braga"
    rdfs:label="Alice Braga"/>

```

EK-2 (Devam) Maris sinema veri modeli RDF

```

<maris:Time_Interval rdf:about="&maris;maris_Instance_63"
    maris:intervals="10:40"
    rdfs:label="10:40"/>
<maris:Time_Interval rdf:about="&maris;maris_Instance_64"
    maris:intervals="13:20"
    rdfs:label="13:20"/>
<maris:Time_Interval rdf:about="&maris;maris_Instance_65"
    maris:intervals="15:55"
    rdfs:label="15:55"/>
<maris:Time_Interval rdf:about="&maris;maris_Instance_66"
    maris:intervals="18:30"
    rdfs:label="18:30"/>
<maris:Time_Interval rdf:about="&maris;maris_Instance_67"
    maris:intervals="21:00"
    rdfs:label="21:00"/>
<maris:Actor rdf:about="&maris;maris_Instance_7"
    maris:name="Salli Richardson"
    rdfs:label="Salli Richardson"/>
<maris:Movie rdf:about="&maris;maris_Instance_71"
    maris:date_launch="25/01/2005"
    maris:language="Turkce"
    maris:name="Ulak"
    maris:type="Dram"
    rdfs:label="Ulak">
    <maris:subject>Zekeriya koy koy dolasip Ulak
    Ibrahimin hikayesini anlatan bir
    gezgindir.</maris:subject>
    <maris:directors
rdf:resource="&maris;maris_Instance_72"/>
    <maris:actors
rdf:resource="&maris;maris_Instance_74"/>
    <maris:actors
rdf:resource="&maris;maris_Instance_75"/>
    <maris:actors
rdf:resource="&maris;maris_Instance_76"/>
    <maris:actors
rdf:resource="&maris;maris_Instance_77"/>
    <maris:actors
rdf:resource="&maris;maris_Instance_78"/>
    <maris:actors
rdf:resource="&maris;maris_Instance_79"/>
    <maris:actors
rdf:resource="&maris;maris_Instance_80"/>
    <maris:actors
rdf:resource="&maris;maris_Instance_81"/>

```


EK-2 (Devam) Maris sinema veri modeli RDF

```

    <maris:actors
rdf:resource="&maris;maris_Instance_82"/>
    <maris:actors
rdf:resource="&maris;maris_Instance_83"/>
    <maris:actors
rdf:resource="&maris;maris_Instance_84"/>
</maris:Movie>
<maris:Director rdf:about="&maris;maris_Instance_72"
    maris:name="Cagan Irmak"
    rdfs:label="Cagan Irmak"/>
<maris:Actor rdf:about="&maris;maris_Instance_74"
    maris:name="Cetin Tekindor"
    rdfs:label="Cetin Tekindor"/>
<maris:Actor rdf:about="&maris;maris_Instance_75"
    maris:name="Humeyra"
    rdfs:label="Humeyra"/>
<maris:Actor rdf:about="&maris;maris_Instance_76"
    maris:name="Yetkin Dikinciler"
    rdfs:label="Yetkin Dikinciler"/>
<maris:Actor rdf:about="&maris;maris_Instance_77"
    maris:name="Serif Sezer"
    rdfs:label="Serif Sezer"/>
<maris:Actor rdf:about="&maris;maris_Instance_78"
    maris:name="Kaya Akkaya"
    rdfs:label="Kaya Akkaya"/>
<maris:Actor rdf:about="&maris;maris_Instance_79"
    maris:name="Melis Birkan"
    rdfs:label="Melis Birkan"/>
<maris:Actor rdf:about="&maris;maris_Instance_8"
    maris:name="Charlie Tahan"
    rdfs:label="Charlie Tahan"/>
<maris:Actor rdf:about="&maris;maris_Instance_80"
    maris:name="Feride Cetin"
    rdfs:label="Feride Cetin"/>
<maris:Actor rdf:about="&maris;maris_Instance_81"
    maris:name="Sener Kokkaya"
    rdfs:label="Sener Kokkaya"/>
<maris:Actor rdf:about="&maris;maris_Instance_82"
    maris:name="Mahir Ipek"
    rdfs:label="Mahir Ipek"/>
<maris:Actor rdf:about="&maris;maris_Instance_83"
    maris:name="Zuhal Gencer Erkaya"
    rdfs:label="Zuhal Gencer Erkaya"/>
<maris:Actor rdf:about="&maris;maris_Instance_84"
    maris:name="Mahmut Gokgoz"

```

EK-2 (Devam) Maris sinema veri modeli RDF

```

        rdfs:label="Mahmut Gokgoz"/>
<maris:Cinema_Movie rdf:about="&maris;maris_Instance_86"
    maris:seance_days="Wednesday"
    maris:seance_times="2150"
    rdfs:label="Ulak-AFM Cepa">
    <maris:cinemas
rdf:resource="&maris;maris_Instance_46"/>
    <maris:movies
rdf:resource="&maris;maris_Instance_71"/>
    <maris:seance_days>Friday</maris:seance_days>
    <maris:seance_days>Monday</maris:seance_days>
    <maris:seance_days>Saturday</maris:seance_days>
    <maris:seance_days>Sunday</maris:seance_days>
    <maris:seance_days>Thursday</maris:seance_days>
    <maris:seance_days>Tuesday</maris:seance_days>
    <maris:seance_times>1110</maris:seance_times>
    <maris:seance_times>1410</maris:seance_times>
    <maris:seance_times>1650</maris:seance_times>
    <maris:seance_times>1920</maris:seance_times>
</maris:Cinema_Movie>
<maris:Director rdf:about="&maris;maris_Instance_92"
    maris:name="Hasan Karacadag"
    rdfs:label="Hasan Karacadag"/>
<maris:Actor rdf:about="&maris;maris_Instance_93"
    maris:name="Ayca Inci"
    rdfs:label="Ayca Inci"/>
<maris:Actor rdf:about="&maris;maris_Instance_94"
    maris:name="Burak Hakki"
    rdfs:label="Burak Hakki"/>
<maris:Actor rdf:about="&maris;maris_Instance_95"
    maris:name="Nazli Ceren Argon"
    rdfs:label="Nazli Ceren Argon"/>
<maris:Actor rdf:about="&maris;maris_Instance_96"
    maris:name="Cem Kurtoglu"
    rdfs:label="Cem Kurtoglu"/>
<maris:Movie rdf:about="&maris;maris_Instance_97"
    maris:date_launch="08/02/2008"
    maris:language="Turkce"
    maris:name="Semum"
    maris:type="Gerilim"
    rdfs:label="Semum">
    <maris:subject>27 yasindaki Canan ve kocasi Volkan
Karaca, yeni aldiklari buyuk eve tasinirlar. Her sey cok
guzel giderken, gunun birinde Canana sebebini
bilmedigimiz garip seyler olmaya baslar.</maris:subject>

```

EK-2 (Devam) Maris sinema veri modeli RDF

```

    <maris:directors
rdf:resource="&maris;maris_Instance_92"/>
    <maris:actors
rdf:resource="&maris;maris_Instance_93"/>
    <maris:actors
rdf:resource="&maris;maris_Instance_94"/>
    <maris:actors
rdf:resource="&maris;maris_Instance_95"/>
    <maris:actors
rdf:resource="&maris;maris_Instance_96"/>
</maris:Movie>
<maris:Time_Interval rdf:about="&maris;maris_Instance_98"
    maris:intervals="10:50"
    rdfs:label="10:50"/>
<maris:Time_Interval rdf:about="&maris;maris_Instance_99"
    maris:intervals="13:40"
    rdfs:label="13:40"/>
</rdf:RDF>

```

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : KARADEMİRCİ, Olgun
 Uyruğu : T.C.
 Doğum tarihi ve yeri : 01.05.1979 Ardahan
 Medeni hali : Bekar
 Telefon : 0 (533) 491 34 36
 Faks :
 e-mail : okarademirci@gmail.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Lisans	Karadeniz Teknik Üniversitesi/ Bilgisayar Mühendisliği	2003
Lise	Bursa Erkek Lisesi	1997

İş Deneyimi

Yıl	Yer	Görev
2003-2004	İnteltek A.Ş., Merkezi Bahis Sis.	Sistem Mühendisi
2004-2005	İnteltek A.Ş., Merkezi Bahis Sis.	Yazılım, Takım Lideri
2005-2006	İnteltek A.Ş., Merkezi Bahis Sis.	Teknik Yönetici
2006-2007	Turkcell İletişim Hiz. A.Ş.	Proje Üyesi
2007-	Smile Holding A.Ş.	Teknik Danışman

Yabancı Dil

İngilizce

Yayınlar

1. Karademirci, O., Hancı A., “Spor Toto Teşkilatı Bayi Uzaktan Eğitimi”, Bilgi Teknolojileri Kongresi III, Pamukkale Üniversitesi, Denizli, 142-144 (2004).