

**T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**ONTOLOJİ TABANLI PERSONEL ALIM SİSTEMİNDE
AĞ SERVİSLERİNİN
TASARLANMASI VE UYGULANMASI**

YÜKSEK LİSANS TEZİ

Bil.Müh. ATILLA ÖZPALA

Enstitü Ana Bilim Dalı : BİLGİSAYAR VE BİLİŞİM MÜH.

Tez Danışmanı : Doç. Dr. Raşit KÖKER

Şubat 2008

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**ONTOLOJİ TABANLI PERSONEL ALIM SİSTEMİNDE
AĞ SERVİSLERİNİN
TASARLANMASI VE UYGULANMASI**

YÜKSEK LİSANS TEZİ

Bil.Müh. ATILLA ÖZPALA

Enstitü Ana Bilim Dalı : BİLGİSAYAR VE BİLİŞİM MÜH.

Bu tez 01/ 02 /2008 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.

Prof.Dr. Ümit KOCABIÇAK
Jüri Başkanı

Doç.Dr. Raşit KÖKER.
Üye

Y.Doç.Dr. A.Fuat BOZ
Üye

ÖNSÖZ

Bu çalışmanın gerçekleşmesinde beni yönlendiren tez danışmanım Doç.Dr.Raşit Köker'e, bana yeni yaklaşımlar sağlayarak yardımcı olan Mehmet Erkanar'a ve Ümit Lütfü Altıntakan'a, her zaman yanımda olan dostum Yusuf Yavuz Ayar'a teşekkürlerimi sunarım

İÇİNDEKİLER

ÖNSÖZ	ii
İÇİNDEKİLER	iii
SİMGELER VE KISALTMALAR LİSTESİ	v
ŞEKİLLER LİSTESİ	vi
TABLolar LİSTESİ	vii
ÖZET	viii
SUMMARY	ix

BÖLÜM 1.

GİRİŞ	1
-------------	---

BÖLÜM 2.

WEB SERVİSLERİ	4
2.1. SOAP	7
2.2. WSDL	9
2.2.1. WSDL bilgi modeli	10
2.3. Web Servis Kayıtçıları	12
2.3.1. UDDI	13
2.3.2. ebXML	16

BÖLÜM 3.

ANLAMSAL WEB VE DİLLERİ	20
3.1. Anlamsal Web Nedir?	20
3.1.1. Ontoloji kavramı	24
3.2. Anlamsal Web Dilleri	25

3.2.1. XML (Extensible Markup Language).....	25
3.2.2. RDF (Resource Description Framework)	27
3.2.3. RDFS (RDF Schema).....	28
3.2.4. OWL (Web Ontology Language).....	30
3.3. Anlamsal Web Servisleri.....	32
3.3.1. OWL-S	32
BÖLÜM 4.	
UYGULAMA	35
4.1. Kullanılan Araçlar	35
4.1.1. Protege.....	35
4.1.2. Jena.....	36
4.1.3. NetBeans	37
4.2. Uygulama Tasarımı	37
4.2.1. Kullanılan ontoloji	38
4.2.2. Sınıflar ve metotları.....	40
4.2.3. Web servisleri.....	42
4.2.4. Web sayfaları	43
4.3. Uygulamanın Gerçekleştirimi	44
BÖLÜM 5.	
SONUÇ VE ÖNERİLER	46
KAYNAKLAR	47
ÖZGEÇMİŞ	51

SİMGELER VE KISALTMALAR LİSTESİ

A2A	: Application-to-Application
B2B	: Business-to-Business
BPEL	: Business Process Execution Language
ebXML	: Electronic business XML
HTTP	: Hypertext Transport Protocol
MEB	: Milli Eğitim Bakanlığı
MERNİS	: Merkezi Nüfus İşleri
OASIS	: Organization for the Advancement of Structured Information Standards
ÖSYM	: Öğrenci Seçme ve Yerleştirme Merkezi
OWL	: Ontology Web Language
OWL-S	: Web Ontology Language for Services
RDF	: Resource Description Framework Schema
RDFS	: Resource Description Framework
SOAP	: Simple Object Access Protocol
UDDI	: Universal Description, Discovery, Integration
URI	: Universal Resource Identifier
W3C	: World Wide Web Consortium
WWW	: World Wide Web
XML	: Extensible Markup Language

ŞEKİLLER LİSTESİ

Şekil 2.1. Web Servis Mimarisi	5
Şekil 2.2. Web Servis İş akışı.....	6
Şekil 2.3. SOAP Mesaj Yapısı	8
Şekil 2.4. WSDL Kavramsal Modeli	9
Şekil 2.5. WSDL Bilgi Modeli (www.xml.org).....	11
Şekil 2.6. UDDI Bilgi Modeli (Building Web Services with Java 2001).....	14
Şekil 2.7. ebXML Bileşenleri.....	17
Şekil 2.8. ebXML İş Süreci.....	19
Şekil 3.1. Anlamsal Web Dilleri Mimarisi (www.w3c.org)	20
Şekil 3.2. Anlamsal Web – Günümüzde Kullanılan Web.....	21
Şekil 3.3. 1989’ da T. Berners-Lee Tarafından Öngörülen Web (T.Berners Lee 1989)..	22
Şekil 3.4. Anlamsal Web Katmanları (T.Berners Lee1998)	23
Şekil 3.5. RDF Grafiksel Gösterimi	28
Şekil 3.6. RDF-S Grafiksel Gösterimi	29
Şekil 3.7. OWL-S Üst Seviye Servis Ontolojisi	34
Şekil 4.1. Personel Alım Ontolojisi.....	38
Şekil 4.2. Personel Alım Ontolojisi Örnek Ağacı	40
Şekil 4.3. Uygulama geliştirme ortamı NetBeans IDE 5.5	41
Şekil 4.4. Kullanılan Uygulama Sunucusu Sun Java System Application Server Platform Edition 9.0_01	43
Şekil 4.5. İş Başvuru Sayfası.....	44

TABLÖLAR LİSTESİ

Tablo 2.1. UDDI İşlemleri	16
---------------------------------	----

ÖZET

Anahtar kelimeler: Anlamsal Web, Ontoloji, Ağ Servisleri, Personel Alımı

İş dünyası her geçen gün süreçlerinin daha büyük bir bölümünü internet üzerine taşımaktadır. Personel alım süreçleri de bu gelişimin paralelinde internette yoğun olarak yer almaktadır. Günümüzde iş dünyasının iş ilanlarını yayınlamak ve iş arayan kişilerin kendilerine en uygun pozisyonu bulmak için en yaygın olarak kullandıkları ortam internettir. Bu nedenle internet üzerindeki insan kaynakları sitelerinin sayısı gün geçtikçe artmaktadır.

İnternetin insanlar tarafından bilgiye ulaşmak için çok etkin bir araç olarak kabul edildiği, hızlı gelişiminden ve içerdiği bilginin sürekli olarak artmasından anlaşılmaktadır. Bu durum bir çok bilgiye ulaşmak için avantajlı olarak değerlendirilebilir, ancak bilginin büyük bir hızla artması doğru bilgiyi bulmayı güçleştirmektedir. Gelecekte daha fazla sorun yaratacağı öngörülen bu durum karşısında başta W3C olmak üzere bir çok kuruluşun ve önde gelen bilişim şirketlerinin önerisiyle “Anlamsal Web” çözümü sunulmuştur.

Bu tezde geliştirilen uygulama ile, firmalar tasarlanmış olan personel alım ontolojisini kullanarak iş ilanı yayımlayabileceklerdir. İş arayan personel adayları, bu iş ilanları arasından kendilerine uygun olan pozisyona başvuracaklardır. T.C. kimlik numarası kullanılarak yapılan bu başvuru sırasında, uygulama devlet kurumlarının sağladığını varsaydığımız web servislerini çağırarak adayın bilgilerini toplayacaktır. Devlet kurumlarından bilgileri toplanan adayların başvurdukları pozisyona uygun oldukları durumda başvurusu kabul edilecektir. Adayın bilgilerinin pozisyona uygun olmadığı durumda adayın uygun olmayan bilgisini içeren bir açıklamayla başvuru olumsuz olarak sonuçlanacaktır.

DESIGN AND IMPLEMENTATION OF WEB SERVICES IN ONTOLOGY BASED RECRUITMENT SYSTEM

SUMMARY

Key Words: Semantic Web, Ontology, Web Service, Recruitment

Nowadays, business world has carried most of the operations and work processes on the Internet. In a parallel manner with this advancement, recruitment processes take place on the web heavily. Internet is the most common environment for companies to publish open positions or search the best suitable jobs for job seekers. Because of this inclination, the number of the deployed human resource sites increases day by day.

It may conclude that people has accepted the internet as the most efficient tool for reaching information because of its fast evolution and the continuously enlargement of information in it. This situation looks like an advantage for reaching huge information but the enlargement of information makes difficult to find the correct information. In the future, the problem will likely to be more critical. To solve this problem which will possibly result in many important problems, especially W3C and many other foundations and leading IT corporations come with the solution “Semantic Web”.

In the thesis, companies can publish their open positions by using the recruitment ontology which was designed. Also job seekers can appeal to suitable positions which are published. Candidates realized this appeal by using their Turkish Republic identity number. During this appeal, application collects information about candidate by using web services which are pretended to be published by government institutions. Candidates, whose information is collected from government institutions, are accepted if his information is suitable for the position. If candidate’s information is not suitable for position, application finalized the appealing process and represent the non suitable information of candidate.

BÖLÜM 1. GİRİŞ

Bir çok iş süreci gibi personel temin süreci de internet üzerinde çok geniş uygulama alanı bulmakta ve bu uygulamalar büyük bir hızla yaygınlaşmaktadır. Günümüzde insan kaynakları yöneticilerinin büyük bir çoğunluğu, internetin çok önemli bir temin kaynağı olduğunu belirtmektedir. Yapılan araştırmalara göre personel alımları büyük oranda internet başvurusu sonucunda yapılmaktadır [1].

İnternette insan kaynakları sitelerinin yaygınlığı özellikle nitelikli personellerin çok geniş bir alanda iş başvurusu yapabilmelerine imkan sağlamıştır. İşverenin personel talebinin çok geniş bir kitleye ulaşması başvurularında aynı nispette artmasına neden olmaktadır[2]. Bu durum işe uygun kişiyi bulma konusunda işverene geniş bir yelpaze sunmasının yanında, seçeneğin çok olduğu bir ortamda en uygun kişiyi bulabilmek gibi bir problem doğurmaktadır. İş verenin açmış olduğu bir kaç personellik iş ilanına binlerce kişi başvurduğunda bu başvurular arasından en uygun kişileri bulabilmek çok zordur. Bundan dolayı işverenler başvuruyu sınırlı sayıda tutabilmek için aslında personelin çalışacağı işle hiç ilgisi olmayan bir çok özellik aradıklarını belirtmekte ki böylece daha az sayıda daha nitelikli kişiler başvuru yapsın. Bu durum ise genelde iki şekilde son bulmaktadır; ya üstün nitelikli personel nitelikleriyle uygun olmayan işten bir süre sonra ayrılmaktadır, yada aslında sahip olmadığı nitelikleri iyi bir sunumla işverene pazarlayan personel gereken verimi sağlayamamaktadır.

Bu sorun karşısında bulunan çözümde işveren gerçekten pozisyon için sadece ihtiyaç duyduğu özellikleri belirtecek ancak şirketin insan kaynakları birimi de bir kişilik pozisyon için binlerce kişiyle görüşmeyecek hatta birçoğunun başvurularını dahi incelemeyecektir [3].

Bütün bu problemler anlamsal web, ontolojiler ve web servisler ile çözebilir. Anlamsal web ile iş verenin talep ettiği özelliklerde ki kadro için gerekli kriterler (eğitim, yaş vb.) insan kaynakları ontolojisi kullanılarak tanımlanabilir ve internette insan kaynakları portallarında sunulur. Benzer şekilde iş arayan kişide kendi özelliklerini insan kaynakları ontolojisini kullanarak belirler. Bunun sonucunda her iki yönlü olarak anlamsal uyuşma sorgulaması yapılarak personele en uygun pozisyonu veya pozisyona en uygun personeli tespit edebilir. Bu durum insan kaynakları personelinin kişilerin özgeçmişlerini tek tek inceleyerek çıkaracakları sonuçtan yada iş arayan birinin tüm ilanları inceleyerek iş bulmalarından çok daha kolay olduğu gibi, daha güvenilir bir sonuç elde edilir.

Buraya kadar ki kısımda bir ontolojiye dayanılarak hazırlanan iş ilanları ve özgeçmişlerin birbiriyle uyuşmalarının sorgulanmasının getirileri anlatıldı. Şimdi ise bu süreci biraz daha canlandırmanın farklı uygulamaları da bu sürece sokmanın katkıları incelenecek. Bu noktada anlamsal web'in yanında işin içine web servisleri de katılacak. Ve bu web servislerinin birbirleriyle anlamsal olarak ilişkileri kurulacak. Buda anlamsal web çalışmaları altında incelenen anlamsal web servislerinin bir örnekleme olarak düşünülebilir.

Özgeçmiş ve iş ilanlarının ontolojiler kullanılarak hazırlanması ve bu iş için kullanılan portallarda saklanması ve sorgulanması tabi ki işimizi çok kolaylaştırır. Ancak bunun bir adım ötesi iş verenin personeli beyanından değil, beyanda belirtilen bilgilerin kaynağından, bulması yada başvuranın beyanını doğrulamasıdır. Örneğin iş veren kendisine uygun eğitim durumunda ki kişileri Milli Eğitim Bakanlığında, bu kişiler arasında yabancı dil bilgisi istediği seviyede olan kişileri ve yabancı dil sınav notlarını ÖSYM'den, bunlar içerisinde beklediği iş tecrübesinde olanları Sosyal Güvenlik ve Çalışma Bakanlığında ve bunlar gibi ihtiyaç duyduğu tüm bilgileri verinin kaynağı olan kurumlardan alabilir.

Bilgiyi almak; listeler halinde veriyi toplayıp, birbiriyle karşılaştırıp, uygun olan adayları tespit etmek değildir. Bu kurumların her birinin sağladığı web servisler, bu web servisleri birbiriyle anlamsal olarak birleştiren ve kaynaştıran bir ontoloji ve bu ontoloji

vasıtasıyla kurumların veri tabanlarını sorgulayan yazılım araçları kullanılarak belirtilen pozisyonun profiline en uygun personel bulunabilir.

Kurgulanan senaryoyu özel sektörün personel alımların da kullanmak için geniş çaplı çalışmalar yapmak gerekir. Çok çeşitli sektörlerin, çok çeşitli olan personel ihtiyaçlarını karşılamak için çok çeşitli kriterleri olabilir. Tüm bu kriterleri belirlemek ve bu kriterleri besleyecek veriyi oluşturmak için insan kaynakları uzmanlığı gerektiren çalışmalar yapmak gerekir. Ancak kamu kurumlarının personel alımında ki kriterleri daha analitik, daha belirgin ve daha az olduğu için bu çalışmada bir kamu kurumunun alım sürecinde ki ihtiyaçları göz önünde bulundurulacaktır. Böylece personel alım sürecinin büyük bir kısmının elektronik ortamda, güvenilir bilgiyle, gerçek hedef kitleyle muhatap olarak, iş gücünü düşürerek ve en doğru personel seçiminin yapılarak sonuçlandırılacağı bir uygulamanın gerçekleştirilebilirliği gösterilecektir.

BÖLÜM 2. WEB SERVİSLERİ

Web servisleri, XML mesajlaşma tabanlı bir uygulama entegrasyon yöntemidir. W3C tarafından yapılan resmi tanımlamaya göre web servisi, bilgisayarlar arasında ağ üzerinden etkileşimi ve uyumluluğu sağlayacak yazılım sistemidir. Günümüzde sistemlerin birbiriyle haberleşmesini gerçeklemek için en çok tercih edilen yöntem web servisidir[4]

Web servisler ağ üzerinde bir URL (Uniform Resource Locator) ile tanımlanan, ortamdaki bağımsız veri iletişimini sağlayan program parçalarıdır. Web servisler bileşen temelli yazılımlardan beklenen en iyi karşılayan yazılım teknolojisidir. Mevcut bileşen temelli yazılımlar (DCOM, RMI, IIOP) gibi nesne modeline özel erişim protokolü kullanmaz. Bunun yerine web servisler genel kabul görmüş olan web erişim protokolü HTTP' yi ve veri formatı olarak da XML' i kullanırlar.

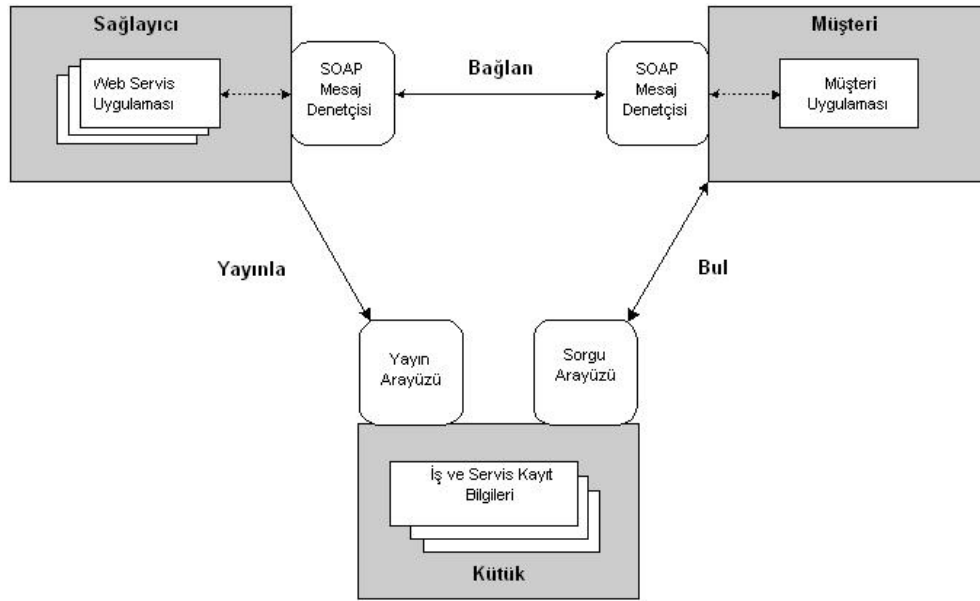
Web servisler için çok çeşitli tanımlamalar yapılmasına rağmen hepsinin ortak paydasına bakarsak web servisler:

- Servis tanımlama dilini kullanarak tanımlanan,
- Servis kayıtçıları aracılığıyla yayınlanan,
- Standart bir yöntemle keşfedilen (discovered),
- Ağ üzerinde herhangi bir uygulama tarafından çağrıla bilinen,
- Diğer servislerle birleştirilebilen,
- Ortamdan ve uygulamadan bağımsız yazılım bileşenleridir.

Web servisleri tanım dokümanında (Web Service Definition Language, WSDL) ve mesaj paketlerinde (SOAP Envelope) XML formatını, transfer protokolü olarak da HTTP' yi kullanmaktadır. Ayrıca web servisleri HTTP protokolünü kullandığı ve

taşıdığı mesajlar düz yazı dosyası şeklinde olduğu için ateş duvarına (Firewall) yakalanmadan istemciler ile haberleşmeyi mümkün kılar[5]

Web servis modeli temel olarak üç öğeden meydana gelir; servis sağlayıcısı, servis kayıtçısı ve servis kullanıcısı. Şekil 2.1'de web servis modelinin grafiksel gösterimi yapılmıştır.



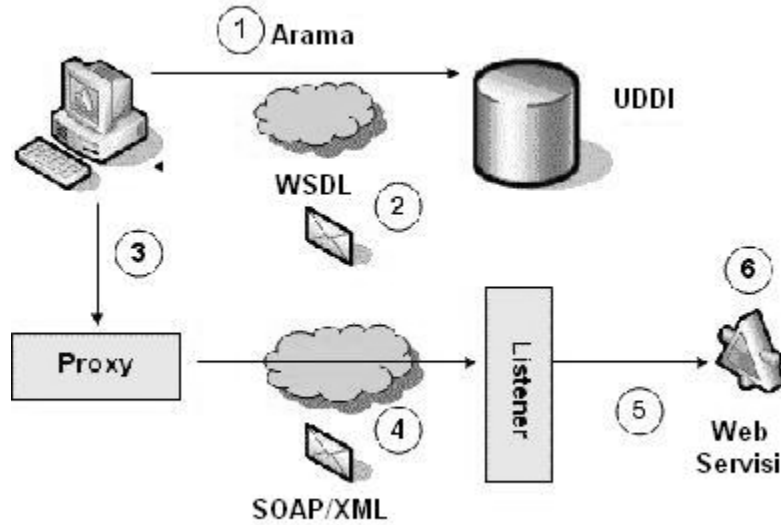
Şekil 2.1. Web Servis Mimarisi

Servis Sağlayıcı (Provider): Servis sağlayıcı istemcilerin sağlayıcıda bulunan servislere erişimini sağlar. Servis sağlayıcı, yayınladığı web servislerin tanımlarını servis kayıt birimine (service registry) kaydederek bu servislerin nasıl çağrılacağını belirtir.

Servis İstemcisi (Consumer) : Servis sağlayıcısında bulunan web servislerini çağırarak kullanan istemci uygulamalardır. Web servisin nasıl çağrılacağını ve ilgili parametrelerini servis kayıt biriminden arayarak bulur ve çağırır.

Servis Kayıt Birimi (Registry) : Servis sağlayıcılarının yayınladıkları web servisi tanımlarını saklar ve aranıp bulunmasını sağlar. Servis sağlayıcıları servis kayıt birimini tarayarak istediği servisler hakkında bilgi alabilir. Servis kayıt birimi her servisin nasıl çağırılacağı konusunda tanım bilgileri içerir.

Şekil 2.2’de belirtilen WSDL, XML formatında, içerisinde web servislerinde kullanılan metotların tanımlandığı dokümandır. Basit Nesne Erişim Protokolü (Simple Object Access Protocol, SOAP) web servisler arasında XML formatında veri değişimi için kullanılmaktadır. UDDI (Universal Description Discovery and Integration), web servislerinin tanımlanması, aranıp bulunabilmesi ve entegre edilebilmesi için web servis ve sağlayıcısı ile ilgili gerekli bilgilerin saklandığı servis kayıtcısıdır[6]



Şekil 2.2 Web Servis İş akışı

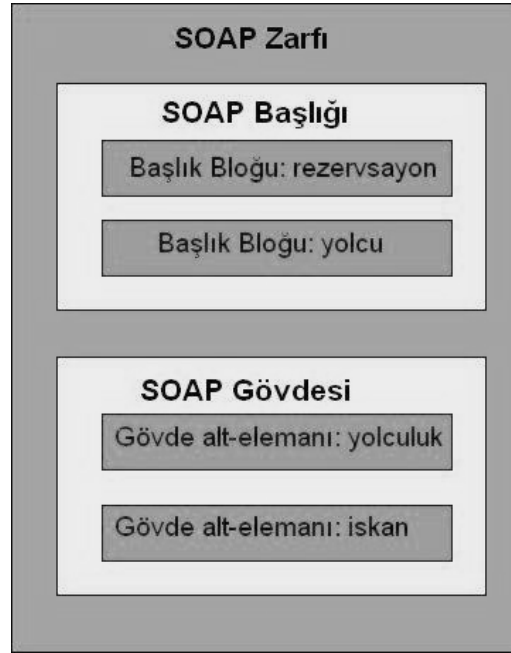
İlerleyen bölümlerde WSDL, SOAP ve UDDI hakkında daha detaylı bilgi verilecektir.

2.1. SOAP

Temel olarak SOAP dağıtık yapıdaki uygulamalar arası (“application to application” A2A) iletişim kurmak için tasarlanmıştır. Özellikle işletmeler arası (B2B) ve ileri uygulama entegrasyonları EAI (Enterprise Application Integration) için kullanılır. Her iki alanda yazılım uygulamalarını tümleştirmeye ve veri paylaşmaya odaklanmıştır. B2B ve EAI uygulamaları ortamdan bağımsız, esnek ve standart yapıda bir protokol kullanmalıdır. EDI ve CORBA gibi, daha önceki B2B ve EAI teknolojilerinin aksine SOAP bahsi geçen ihtiyaçları karşılamaktadır ve birçok kurumsal yazılım satıcısı ve önde gelen standart organizasyonları tarafından kabul görmüştür (W3C, WS-I, OASIS, vb.)[7]

Aslında SOAP kendi kuralları olan XML formatı da veri paketlerinden daha fazlası değildir. SOAP’ ın hedefi , ağ üzerinde dağıtık yapıda ki uygulamalar arasında yazılım donanım farkı gözetmeksizin veri değişimini mümkün kılmaktadır.

SOAP XML dokümanı örneği; SOAP mesajı yada SOAP zarfı olarak adlandırılır.(Şekil 2.3) SOAP mesajı genellikle network protokolleri ile taşınır. Örneğin SOAP mesajlarını uygulamalar arasında kullanmanın en yaygın yolu HTTP üzerinden web gezgini ile HTML web uygulamalarına erişmektir. Burada ki en temel fark SOAP mesajlarını gezgin üzerinde HTML sayfaları görüntülediğiniz gibi görüntüleyemezsiniz. SOAP mesajları ağ üzerinde ki uygulamalar tarafından kullanılır, kişi için kullanılabilir değildir. HTTP, SOAP mesajlarının taşınabilmesi için uygun bir yöntemdir ancak tek yöntem değildir. Bunun yanında mesajlar SMTP (Simple Mail Transfer Protocol) kullanılarak e-posta yoluyla veya FTP ve TCP/IP protokolleri kullanılarak da gönderilebilir.



Şekil 2.3 SOAP Mesaj Yapısı

SOAP, mesajların herhangi bir programlama dilinden ve platformdan bağımsız olarak nasıl yapılandıracağını ve işleteceğini tanımlar. Böylece farklı dillerde yazılmış ve farklı işletim sistemleri üzerinde çalışan programlar arasında ilişki kurulmasına imkan sağlar.

Uygulamalar arasında entegrasyon kurmak için farklı standartlarda bulunmaktadır. Ancak SOAP çok kısa bir sürede geniş alanda kabul görmüş ve yaygınlaşmıştır. Günümüzde SOAP mesajlaşma formatı; Microsoft, IBM, Sun Microsystems, BEA, HP, Oracle, SAP gibi bilişim dünyasının devleri tarafından benimsenmiştir. SOAP' ın bu başarısının temel nedeni XML tabanlı olmasıdır. Bu sayede çok esnek bir yapıya sahiptir. Ayrıca ateş duvarlarına karşı bağımsızlığı olan HTTP gibi tüm dünya tarafından kabul görmüş iletişim protokolünü kullanması da hızlı yaygınlaşmasında büyük etken olmuştur.

2.2. WSDL

Herhangi bir web servisini kullanabilmek için mesajın yapısını, hangi protokolün kullanılacağını ve web servis sağlayıcısının adresini önceden biliyor olmamız gerekir. Ve bu bilgiler herhangi bir ortamda bir web sayfasında olmasından ziyade belirli bir mesajlaşma standardı oluşturulması gerekmektedir. Bu ihtiyaç nedeniyle web servis tanımlama dili WSDL web servis kullanıcıları tarafından benimsenmiştir[8]

Servis tanımlama SOA' nın (Servise Dayalı Mimari) en önemli konularından biridir. Yayınlama, bulma ve bağlama işlemlerinin her biri içinde servis tanımı yapılır. Servis sağlayıcı, servisin tanımını bir veya daha fazla servis kütüğüne kaydeder, servisin tanımı kütüklerden yayınlanır ve servis istemcileri servisleri bu tanımlamalara göre arar, bulur ve web servisini kullanmak için gereken tüm bilgiyi buradan alır. Şekil 2.4' de WSDL' in kavramsal yapısı gösterilmiştir[9].



Şekil 2.4 WSDL Kavramsal Modeli

WSDL web servisleri teknik olarak çağırma için önerilen standart bir söz dizimidir. WSDL, “WSDL Schema” tanımına uygun bir XML dokümanıdır. WSDL web servisin üç temel özelliğini tanımlar.

WSDL dokümanının cevap verdiği sorular aşağıdakilerdir:

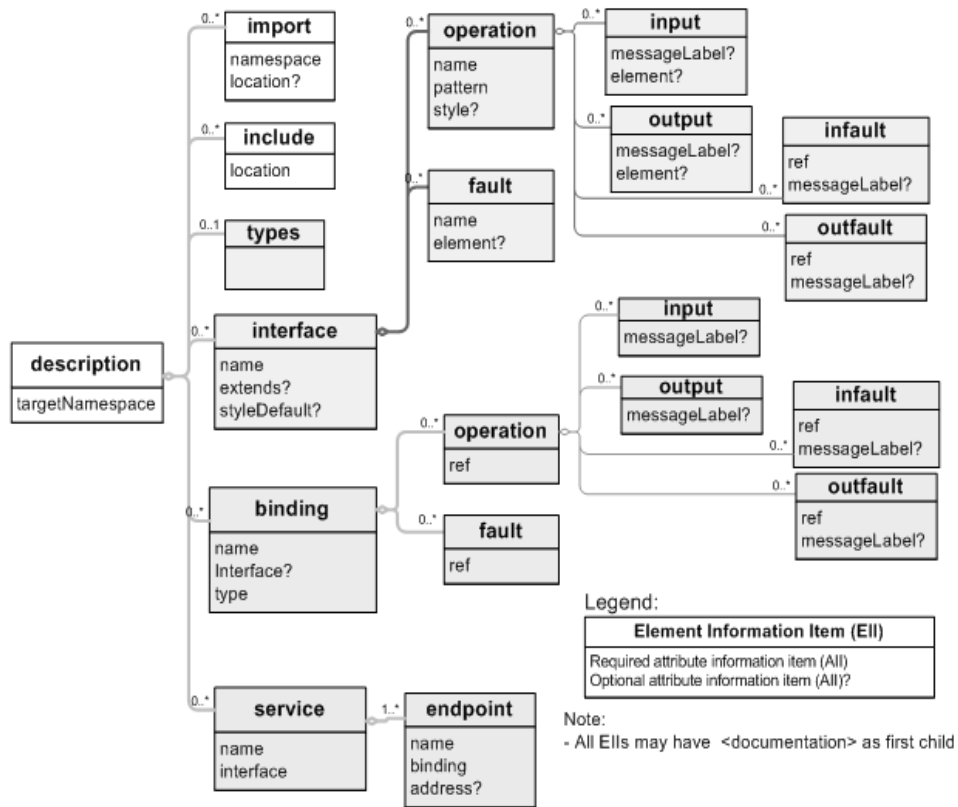
- Web servis nedir: servisin sağladığı işlemler (metotlar) nelerdir.
- Servise nasıl erişilir: servisin işlemlerine ulaşmak için gereken protokoller ve veri formatları nelerdir.
- Servis nereye yerleştirilmiştir: ağ üzerinde adres ile ilgili detaylar belirtilir, (URL vb.)[5].

2.2.1. WSDL bilgi modeli

WSDL bilgi modeli, soyut özelliklerle bu özelliklerin somut uyarlamalarının ayrımından kaynaklanan avantajları çok iyi kullanır . Bu ayrım servis ara yüzü tanımıyla (soyut ara yüz) servis uyarlama tanımının (somut ara yüz) ayrıştırılmasının yansımasıdır.

WSDL çıkış noktasının yetenekleri “<portType>” elemanı ile sunulan soyut bir ara yüz vasıtasıyla tanımlanır. WSDL bağlantı yöntemini “<binding>” elemanı ile sunar. Herhangi bir web servis uygulamasında hangi mesajlaşma, veri şifreleme ve iletişim yönteminin eşleneceğinin soyut tanımı “<binding>” elemanı ile yapılır. “<binding>” elemanı uygulamanın erişilebileceği bir adresle eşlendiğinde, soyut çıkış noktası, servis istemcilerinin servisi çağırabileceği somut bir çıkış noktasına dönüşür. Bu eşleme WSDL “<port>” elemanı ile yapılır.

Soyut ara yüz bir çok işlemi destekler. Bu işlemler etkileşimli çalışma şablonu sunan mesaj kümeleriyle tanımlanır. Mesajlara ve işlemlere ait soyut kavramların somut karşılıkları “<binding>” elemanı ile belirtilir(Şekil 2.5)[10].



Şekil 2.5 WSDL Bilgi Modeli (www.xml.org)

XML' in bütün güçlü uygulamaları gibi WSDL şemasının da üst seviyeli elemanları vardır. Bunların en temel olanları aşağıda sıralanmıştır:

portType : Soyut metot tanımlamasının yapıldığı her bir alt işlem elemanı için web servise ait soyut bir ara yüz tanımlar (Java ara yüz tanımı gibi).

message : Metotların ve işlem elemanlarını başvurduğu parametrelerin tanımlandığı elemandır.

Types : Bütün web servislerin içerisinde kullanılan, tüm mesajların başvurduğu veri tiplerinin tanımlamasını yapar.

Binding : Soyut ara yüzde ki (“<portType>”) elemanların, belirtilen veri formatları ve protokollerin birleşimine nasıl dönüştürüleceğini ve bu dönüşümün somut olarak nasıl sunulacağını detaylarını içerir.

Port : Binding ve web adresinden oluşan servis noktasını tanımlar. Web adresi servisin çalıştırılacağı URL’ dir.

Service : Tüm portların tanımlandığı elemandır. Örneğin çok adımlı bir işlemde portların birleştirildiği yerdir.

Sonuç olarak “<portType>” elemanı web servisin, ne olduğunu, “<binding>” elemanı nasıl olduğunu ve “<port>” ile “<service>” elemanı nerede olduğunu tanımlar [6].

2.3. Web Servis Kayıtçıları

Web servislerini kayıt altında tutma, keşfetme ve yönetebilme servis odaklı mimari (SOA) ile geliştirilen uygulamaların getirdiği temel gereksinimlerdir. Bu gereksinim SOA’ nın ilk safhalarında az sayıda servisle çalışılırken tam olarak anlaşılammıştır. Ancak büyük iş süreçlerinin çok sayıda web servisi tarafından desteklenmesi gerektiğinde ve bu servislerin sayısı yüzler seviyesine yükseldiğinde, servislere ulaşmak ve onları yönetmek kritik bir konu haline gelmiştir. Servis sağlayıcısı ve servis kullanıcısı ile birlikte üç temel bileşenden biri olan servis kayıtçıları bu noktada devreye girer. Bu nedenle servis kayıtçıları SOA için anahtar bir altyapı bilişenidir.

Organizasyonlar arasında birlikte çalışabilirliği sağlayabilmek için web servis kayıtçılarının da diğer web servis bileşenleri gibi bir takım standartlar üzerine oturması gerekir. İlk nesil web servislerin çoğunluğu servis açıklamalarına odaklanan UDDI standartlarına dayanmaktadır (örneğin: WSDL açıklamaları). Bu temel bir fonksiyondur, ancak şuan SOA projeleri WSDL ile tanımlanamayacak, servisle bağlantılı çok daha geniş bir meta veriden oluşmaktadır. Bunlar içerisinde XML şemalar, BPEL tanımları, XSLT dönüşümleri vb. meta veriler vardır.

Bu alanda ki diğerk bir standart da ebXML' dir. ebXML, web servis kayıt fonksiyonlarını desteklemenin yanında her çeşit servis meta verisinin organizasyonu, saklanması ve kontrolü için fonksiyonlar sağlar. Fonksiyonlar bu özellikleriyle birleştirilmiş web servislerinin, havuzlar arası varlık yönetimini sağlar. Web servis kayıtçılarının ve havuzlarının birleştirilmesi servislerin bulunabilmesini kolaylaştırdığı gibi büyük ölçütlü SOA projelerinin yaşam döngüsünün yönetimi için esnek bir altyapıda sağlar [7].

2.3.1. UDDI

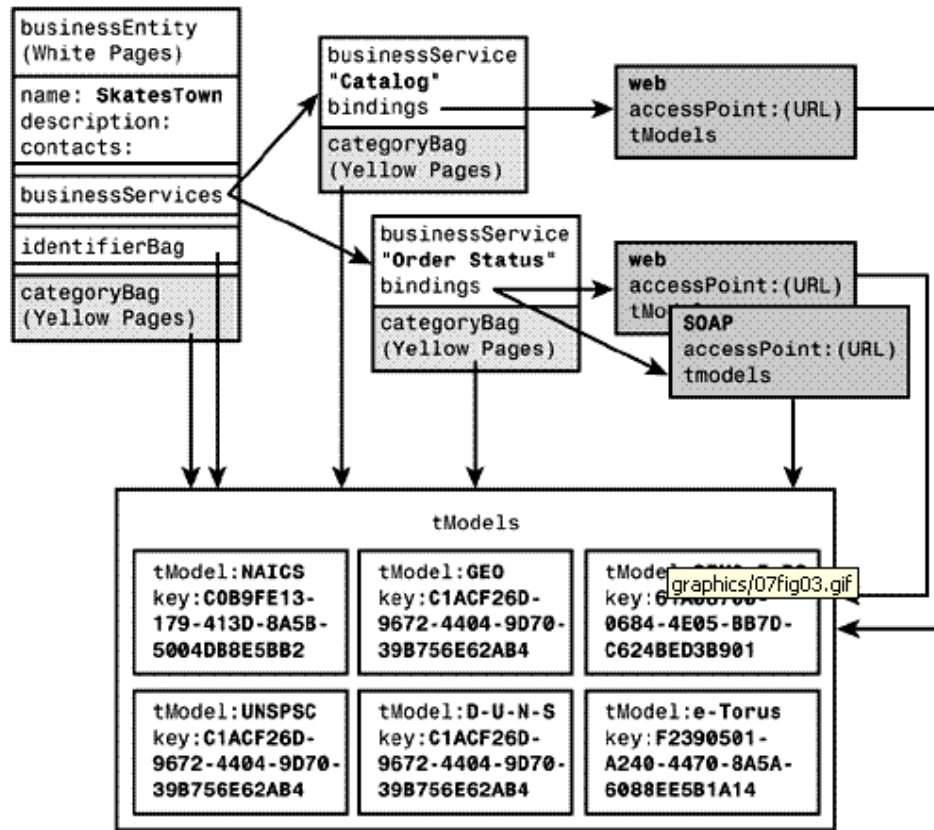
UDDI, web servisler için adres defteri hizmeti vermek üzere tasarlanmıştır. Servis kayıtlarının nasıl bina edileceğinin, her birinin nasıl bulunacağıının ve internet üzerinde etkileşim kurmak için gereken tanımlamanın standart yolunu gösteren servis kayıt mimarisidir. Tasarımda sağlayıcı olan işletmenin web adresi, servisin sundukları için detaylı bir tanım dokümanı, servislerle iletişim kurmak için belirlenen taşıma yöntemi ve veri değışimi için gereken giriş-çıkış parametreleri belirtilir. UDDI başlangıçta IBM, Microsoft ve Ariba ortaklığı tarafından desteklendi, daha sonra başka şirketlerde bu topluluğa katıldı [11].

UDDI işletmeler ve organizasyonların kendileri hakkındaki genel bilgileri ve sağladıkları servilerle ilgili detaylı bilgiyi paylaşmalarını sağlar. Ayrıca işletmelerin, organizasyonların ve endüstriyel grupların sağladıkları servislerin standartları, kısaltmaları, tipleri ve servisler için tahsis edilmiş kimlik bilgileri UDDI kütükleri içerisinde bulunur. Gerçekte UDDI iki farklı kütük önerir: iş kütüğü ve tip referans kütüğü. Buna bağılı olarak iki temel veri yapısı vardır: businessEntity ve tModel.

BusinessEntity veri yapısı kavramsal olarak üçe ayrılır,

- Beyaz sayfalar, varlıklar hakkındaki genel iletişim bilgilerini içerir.
- Sarı sayfalar, servislerin sunduğı varlıkların tipleri ve konumları hakkında sınıflandırma bilgisi sağlar.
- Yeşil sayfalar, servisin nasıl çağırılacağı hakkında detaylı bilgi verir.

Dikkat edilmesi gereken konu bahsi geçen ayırım sadece kavramsaldir. Ayırım her ne kadar kavramsalda olsa bir ögenin beyaz sayfalarında ki sınıflandırma bilgileri sarı sayfalardaki farklı bölümleri referans gösterir. Yine bir ögenin iş servis girişleri (business service entries) yeşil sayfalara başvuran geliştirim bilgilerini içerir. Bilgi modeli yapısının detayları Şekil 2.6' da belirtilmiştir.



Şekil 2.6 UDDI Bilgi Modeli (Building Web Services with Java 2001)

UDDI bilgi modelinde ki diğer bir önemli veri yapısı tModel elemanıdır. Web servislerin çağırılma detaylarının tanımlandığı yeşil sayfalarda, servis tipi ve diğer özet bilgileri için işe özel bağlantı detayları tanımlanır. Bu yeniden kullanılabilir özet bilgiler, teknoloji modelleridir. UDDI'da teknoloji modelleri ile ilgili olan veri yapısı tModel elemanıdır. Endüstri gruplarının, bir takım standartların veya her hangi bir işletmenin

servislerinin kullanılması için gereken yeniden kullanılabilir özet bilgileri tModel elemanın da tanımlanır. tModel web servisler için imza niteliğindedir.

UDDI, web servislerini yayınlamak (publication API) ve sorgulamak (inquiry API) için ara yüzler sağlar. Bu ara yüzler HTTP ve HTTPS üzerinden iletile bilinen ve tüm UDDI kütükleri tarafından anlaşılan 20 SOAP mesajı tanımlamaktadır. Ara yüzlerde kullanılan SOAP mesajların gövdeleri UDDI bildiriminde tanımlanan XML yapılarındadır.

Yayın arayüzü; kurumların bilgilerini, işlerini, uyarlamalarını veya servis tipi özelliklerini yayınladıkları, kimlik doğrulaması yapılarak kullanılan süreçlerden oluşur. UDDI içerisinde bir kimlik doğrulama metodu barındırmaz. Bunun yerine her işletmen kendi kimlik doğrulama işleyişini kendisi sağlar.

Sorgu ara yüzü; kimlik doğrulaması gerektirmeyen, UDDI kütüğü dışındaki kullanıcıların bilgilere ulaşmasına olanak sağlayan, umumi işlemlerden oluşmaktadır. Sorgular iki farklı yöntemi izler; genel bilgileri bulmak için göz atma yöntemi (“find” işlemleri) ve detaylı bilgileri bulmak için sondaj yöntemi (“get_Detail” işlemleri) kullanılır.

UDDI içerisindeki dört yapı tipine karşılık (businessEntity, businessService, bindingTemplate, tModel), dört kayıt işlemi (save_business, save_service, save_binding, save_tModel), dört silme işlemi (delete_business, delete_service, delete_binding, delete_tModel), dört arama işlemi (find_business, find_service, find_binding, find_tModel) ve dört detay arama işlemi (get_businessDetail, get_serviceDetail, get_bindingDetail, get_tModelDetail). (Tablo 2.1)[6]

Tablo 2.1 UDDI İşlemleri

	İş (Business)	Servis (Service)	Bağlayıcı (Binding)	tModel
Kaydet/Güncelle	save_business	save_service	save_binding	save_tModel
Silme	delete_business	delete_service	delete_binding	delete_tModel
Bulma	find_business	find_service	find_binding	find_tModel
Detaylar	get_businessDetail	get_serviceDetail	get_bindingDetail	get_tModelDetail

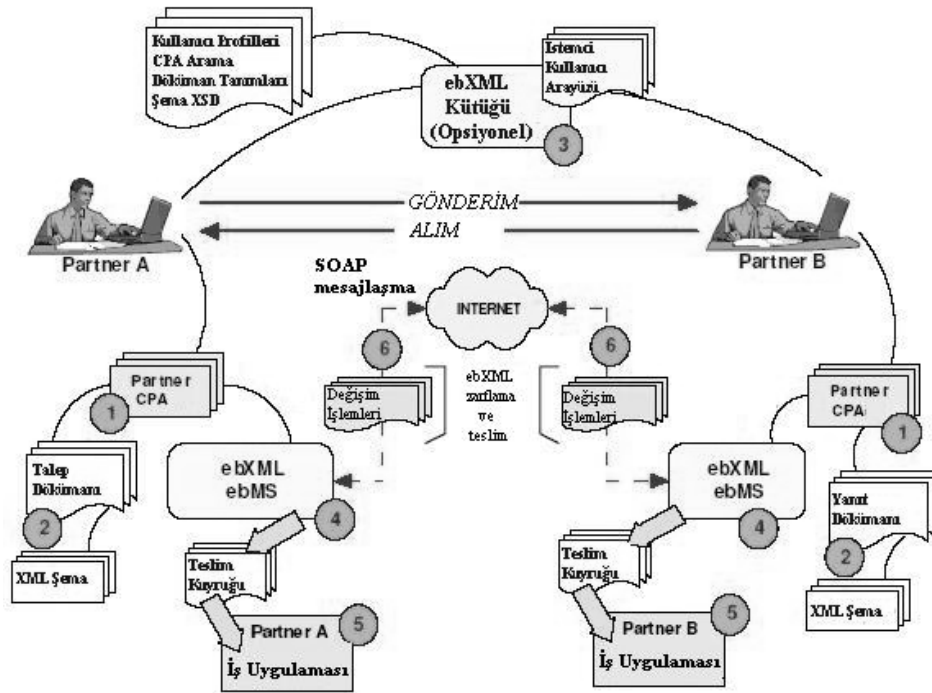
Kayıt işlemleri yeni bir giriş yapmak için yada mevcut bir kaydı değiştirmek için kullanılır. Yeni bir giriş yaratılırken girişin kimlik anahtar parametresi işletmen tarafından tahsis edilir. Daha sonra mevcut bir kaydı güncellemek veya silmek istendiğinde işletmen tarafından tahsis edilen kimlik anahtar parametresi kullanılır.

Arama işlemi bir grup parametre kullanılarak belirtilen kriterlere uygun bilgileri içeren kayıt listesini getirir. Detay arama işlemi ise tanımlayıcı anahtar parametresi verilen bir ögenin detay bilgileri getirir.

2.3.2. ebXML

ebXML'in hedefi elektronik verinin işletmeler arası (B2B) ve işletmelerle müşteriler (B2C) arasında iletimini sağlamaktır. OASIS ve UN/CEFACT ortak girişimleriyle ortaya çıkarılmıştır[12].

ebXML elektronik ortamda iş süreçlerini modüler ve aynı zaman da birlikte işleyebilir bir yapıda gerçekleştirebilmemizi sağlayan kurallar kümesidir. ebXML'in vizyonu, herhangi bir coğrafi bölgede, her büyüklükteki kuruluşlar için küresel bir elektronik pazar oluşturmak ve işletmelerin birbirleriyle XML tabanlı mesajlar yoluyla iletişim kurmasını sağlamaktır [13].



Şekil 2.7 ebXML Bileşenleri

ebXML anlamak için öncelikle onun alfabesi olarak da niteleyebileceğimiz terminolojisini bilmek gerekir. Şimdi bu terimlerden temel olanları tanıyalım (Şekil 2.7);

Kütük (Registry): ebXML in çalışmasını sağlamak için gereken çeşitli veriyi barındıran merkezi bir sunucudur. Bir işletme başka bir işletmeyle ebXML yoluyla bağlantı kurmak istediğinde kendine uygun işletmeyi bulmak için kütüğü sorgular.

İş Süreçleri (Business Processes): Bir işletmenin çalışma sahalarının ve kendisine ortak aradığı iş süreçlerinin tanımlanmasıdır. İş süreçlerinin esas tanımlandığı yer bir XML şeması olan İş Süreci Tanım Şemasıdır (Business Process Specification Schema) ancak UML ile de modellenebilir.

Ortaklık Protokol Tanıtımı (Collaboration Protocol Profile, CPP): ebXML iş süreçlerini kullanmak isteyen işletmelerin elektronik ortamda iletişim kurmak için gerekli detayları

tanımladığı DTD (document type definition) ve XML şemalarını sağlar. CPP işletmenin iş süreçlerini ve sağladığı servis arayüzlerini tanımlar.

İşletme Servis Ara yüzü (Business Servis Interface): İşletme iş süreçlerinde gerekli olan işlemleri gerçekleştirmek için servis ara yüzünü kullanır. Ayrıca servis ara yüzü işletmenin desteklediği bir takım mesajlar ve bu mesajların üzerinde dolaşabildiği iletişim protokolleri içerir.

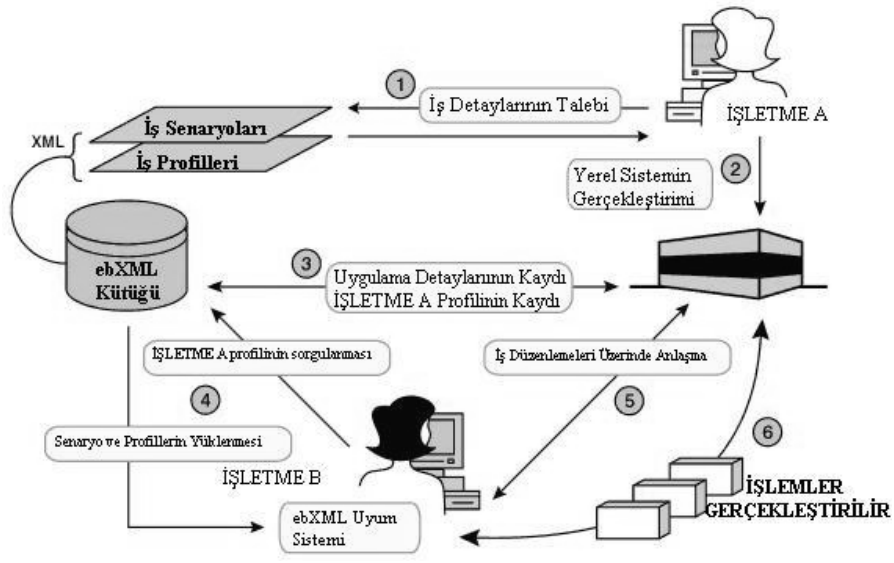
İş Mesajları (Business Messages): Bir iş sürecinin bilgi iletimini sağlayan parçasıdır. Bir mesaj bir çok katmandan oluşabilir. En dış katmanda geçerli bir iletişim protokolü kullanılmadıkça (HTTP veya SMTP gibi). SOAP, mesajları zarflayan ebXML'in önerdiği bir protokolüdür. Diğer katmanlar şifreleme veya kimlik denetimi için kullanılabilir.

Çekirdek Kütüphane (Core Library): Büyük ebXML elemanlarında kullanılan standart parçalar kümesidir. Örneğin çekirdek işlemlere iş süreçlerinden başvurulabilir. Çekirdek kütüphane ebXML girişimcileri tarafından sağlanmaktadır.

Ortaklık Protokolü Uzlaşması (Collaboration Protokol Agreement, CPA): İki yada daha fazla işletmenin CPP' lerinden otomatik olarak türeyen anlaşmalardır. Örneğin bir işletmenin CPP'si "Ben X işini yapabiliyorum" diyorsa CPA "Biz X işini birlikte yapabiliriz" der.

SOAP: Dağıtık ortamlarda veri değişimini sağlamak için kullanılan, ebXML girişimcileri tarafından da desteklenen bir W3C protokolüdür. SOAP, ebXML mesajların içeriğinde ne olduğunu ve nasıl işlete bilineceğini tanımlayan bir zarf oluşturulmasını sağlar [14].

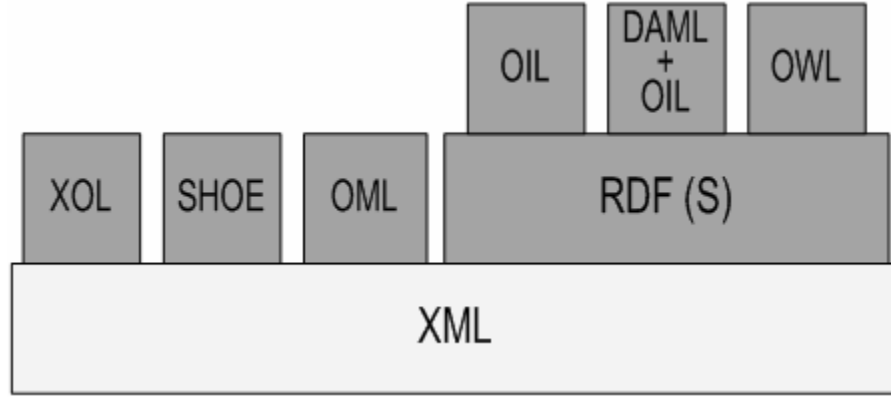
Yukarıda anlatılan öğeleri kullanılarak sürdürülen ebXML iş süreci Şekil 2.8'de grafiksel olarak gösterilmiştir.



Şekil 2.8 ebXML İş Süreci

BÖLÜM 3. ANLAMSAL WEB VE DİLLERİ

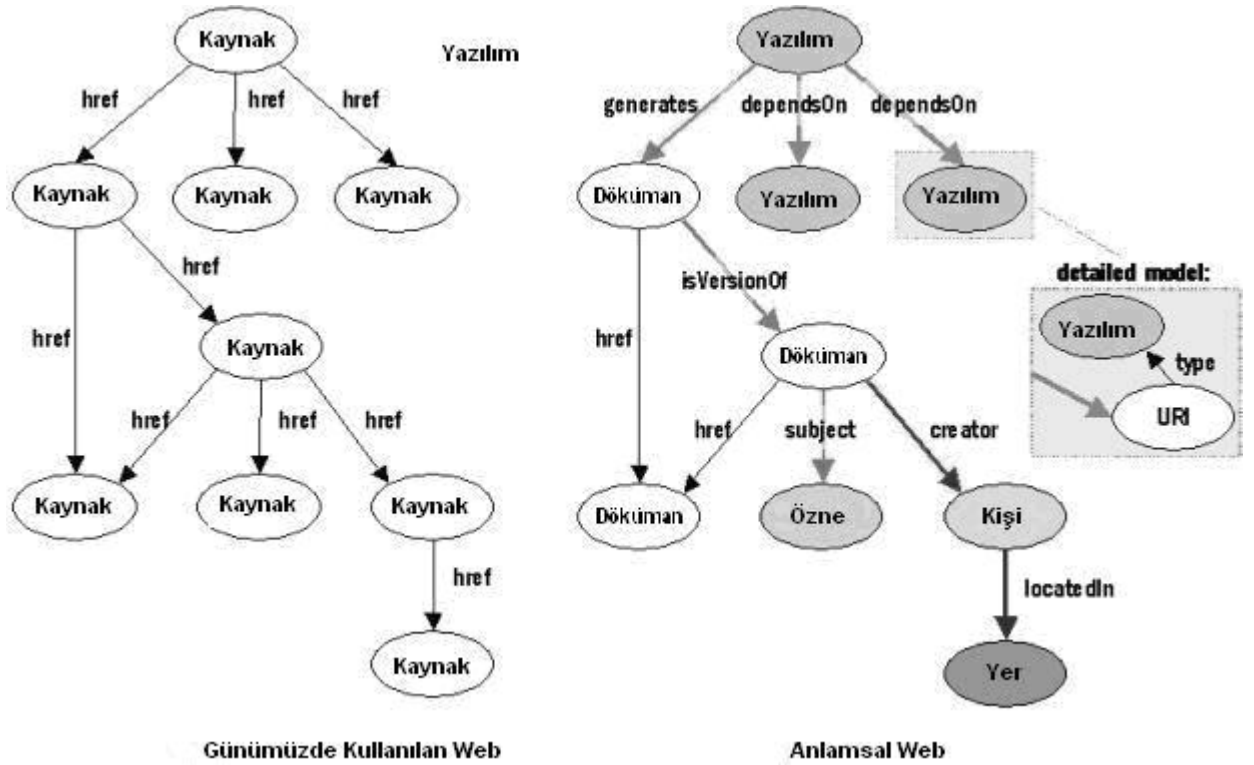
Anlamsal web dilleri (Şekil 3.1) ontolojilerin ve ontolojilerle web ortamındaki bilgi içeren nesnelerin (kaynakların) tanımlanmasını sağlar.



Şekil 3.1 Anlamsal Web Dilleri Mimarisi (www.w3c.org)

3.1. Anlamsal Web Nedir?

Anlamsal web kavramı, bugünkü web' in temelini oluşturan URI, HTTP ve HTML gibi yapılarını tasarlayan ve bulan kişi olan Tim Berners-Lee tarafından öne sürülmüş ve mevcut web ortamının geliştirilerek tam potansiyel kullanımı için web' in gelecek adımı olarak düşünülmektedir. Günümüzde kullandığımız web ve anlamsal web Şekil 3.2' de grafiksel olarak karşılaştırılmıştır.

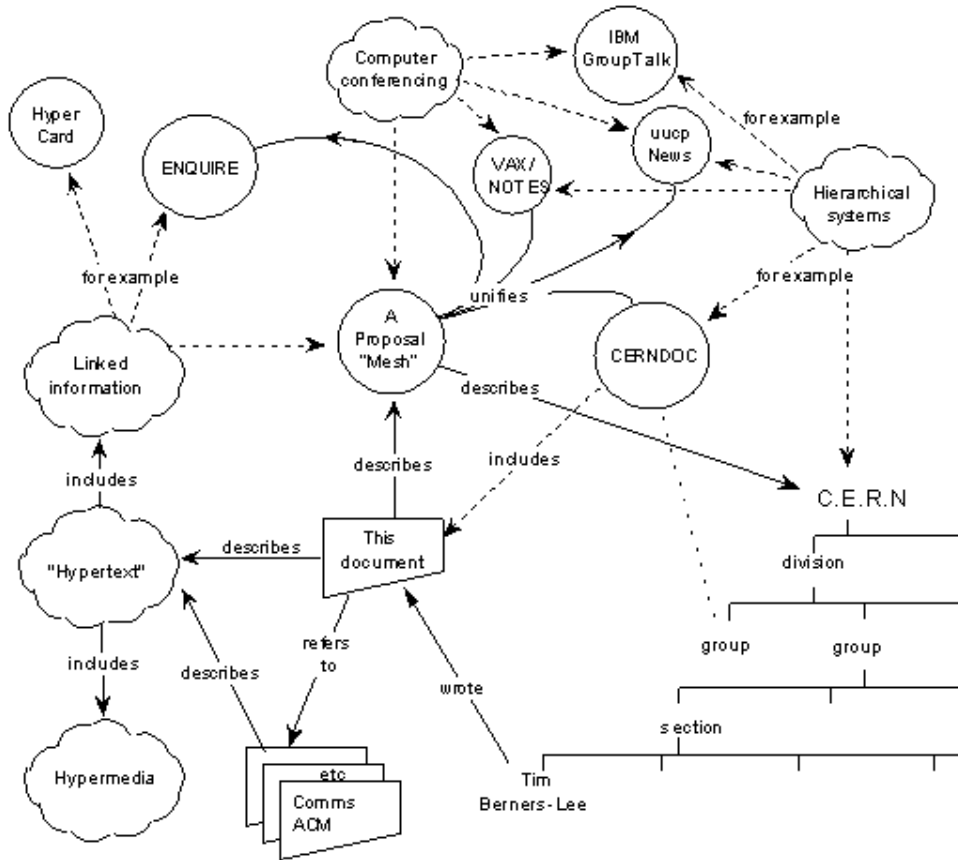


Şekil 3.2 Anlamsal Web – Günümüzde Kullanılan Web

Anlamsal web yeni ve ayrı bir web olmayıp, bilgilere iyi tanımlanmış anlamların verildiği, bilgisayarların ve insanların birlikte çalışmalarına imkan veren bugünkü web' in bir uzantısıdır [15].

Anlamsal web' deki temel amaç iyi tanımlanmış ve aralarında bağlantı kurulmuş olan bilgilerin ve servislerin web ortamında kolay bir şekilde bilgisayarca-okunabilir ve bilgisayarca-anlaşılabilir olmasını sağlayacak standartların ve teknolojilerin geliştirilmesidir.

Web' in ileriye yönelik konumlandırılması Tim Berners-Lee tarafından iki ana vizyon cümlesiyle şekillendirilmiştir; “İlk olarak web, daha fazla birlikteliğin sağlanacağı bir ortam haline gelecektir, ikinci olarak da işlemlerin bilgisayarlar tarafından yürütülmesi ile web daha anlaşılır olacaktır” (Şekil 3.3) [4].

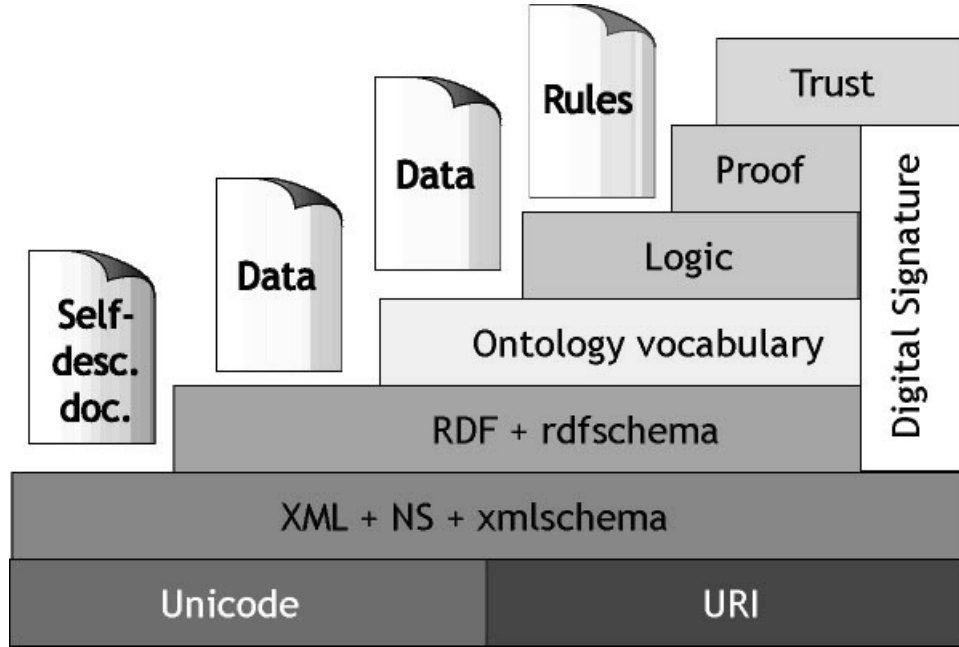


Şekil 3.3 1989' da Tim Berners-Lee Tarafından Öngörülen Web (T.Berners Lee 1989)

Tim Berners-Lee'nin bu vizyonu, HTML verilerinin ve aralarındaki ilişkilerin bilgisayarlar tarafından anlaşılır hale getirilmesi için web' in yeniden tasarlanması anlamına gelmektedir ve Anlamsal Web (Semantic Web) olarak adlandırılmaktadır. Anlamsal Web, günümüzde kullanılan web' in genişletilmiş halidir ve hangi bilginin anlamının daha iyi olduğu, bilgisayarlar tarafından daha iyi anlaşıldığı ve insanların daha çok işine yaradığı hakkındadır

Anlamsal Web' in temel amacı, iyi tanımlanmış ve ilişkilendirilmiş olan bilgilerin ve servislerin, web ortamında bilgisayarlarca okunabilir ve anlaşılabilir olmasını sağlayacak standartların ve teknolojilerin geliştirilmesidir. Anlamsal Web sadece web sayfalarındaki bilgileri kullanan bir insan-bilgisayar ilişkisi olmayıp, verinin bulunduğu diğer ortamlar olan; veri tabanları, servisler, programlar, çeşitli sensörler, kişisel araçlar ve hatta ev

içindeki uygulamalara ait verilerin de işlenebileceği bir ortamdır. W3C tarafından tanımlanan anlamsal web katmanları Şekil 3.4’de gösterilmiştir.



Şekil 3.4 Anlamsal Web Katmanları (T.Berners Lee1998)

Belirtilen ihtiyaçlar, içeriğinde verinin yapısını barındıran ve bilgiyi, veri alış verişi yapan iki taraf içinde anlamlı kılan XML ve XML Şemalarda belirli bir seviyeye kadar karşılamaktadır. Ancak bilginin transferi için sadece transfer edilecek yada paylaşılacak bilginin yapısını bilmemiz yeterli değildir.

Gerçek bir veri paylaşımı yapabilmek için veri içerisinde kullanılan bilgilerin sadece yapısal olarak değil anlamsal olarak eşlemesinin yapılması gerekmektedir ve aynı zamanda bu eşlemenin güncel tutulması gerekmektedir. Anlamsal eşlemeden kasıt paylaşılan bilginin ne anlama geldiğinin bilinmesi ve verinin alanları arasındaki ilişkilerin tanımlanmasıdır [4].

Elektronik ticaret üzerinden bir örnek vermek istersek, aynı türde ürünleri farklı toptancılardan tedarik eden bir satıcı aldığı ürünlerin farklı kodlarla sistemine

aktarmasına karşın bu ürünlerin aynı ürün olduğunu bilmesi gerekmektedir. Bunun içinde toptancılar ve satıcılar arasında hareket gören ürün ve ürün grupları için bilgilerin anlamsal olarak eşelemesinin yapılacağı bir ontoloji oluşturulması gerekmektedir. Böylece herkes kendi sisteminde bilgisini kendi istediği şekilde saklar ve paylaşma ortamında ortak bir dil kullanırlar.

3.1.1. Ontoloji kavramı

Ontoloji doğayla varlıklar arasındaki ilişkilerle ilgilenen felsefe dalıdır. Doğadaki varlıkların çeşitleri ve sınıflandırılmasıyla ilgilenir. Ontolojinin temel manası yukarıda ki tanımlandığı gibi olsa da son on yıl içerisinde ontoloji kavramı bilgi teknolojileri alanında ayrı bir anlam kazanmıştır [16].

Bilgi teknolojileri alanında ontoloji belirli bir bilgi alanında (Domain) ki varlıkların, kavramsal olarak anlamlarını ve aralarındaki ilişkileri içeren bir sözlük olarak tanımlanabilir. Burada bilgi alanından kasıt ortak kavramların kullanıldığı süreçler veya alanlardır. Örneğin sağlık, araba sektörü, finansal planlama, jeoloji, turizm, insan kaynakları yönetimi, kimya vb.

Bir bilgi alanını tanımlamak o bilgi alanında konuşulan ve yazılan sözcüklerin tanımlanmasıdır. Örneğin otomobil servis sektörü tanımlanırken şu konuların tanımlanması gerekir:

- Arabaların çeşitleri (sedanlar, hatchbackler, spor arabalar, lüks sınıf arabalar, yerli ve ithal arabalar)
- Motor çeşitleri (Yakıt türüne göre; benzinli, dizel, elektrik motorlu, hibrit vb.)
- Üreticiler (Mercedes, BMW, Ford, Nissan, Honda, Tofaş vb.)
- Diğer özellikleri (vites sistemi, soğutma sistemi, elektrik sistemi, fren sistemi,)

Arabaların özellikleri dışında bilinmesi gereken bir başka konuda tamir işleminin süreçlerinin tanımlanmasıdır. Hangi arıza belirtilerinin olduğu, teşhis türleri, tamirat işlemleri ve bunu yapacak uzman personel, hangi yedek parçaların bulunduğu, hangi

aletlerin kullanılacağı, tamirat işlemlerinin maliyetinin ne olduğu, ne kadar zaman alacağı vb. iş adımları da bilgi alanında tanımlanmalıdır.

Bunlarla beraber bir de kurallar tanımlanmalıdır. Örneğin bir aracın belirtileri karşılığında hangi arızanın teşhisi ve bu teşhis karşılığında nasıl bir tamirat işlemi hangi uzman personelin atanacağı ve bu tamirat yapılırken hangi takımların ve yedek parçaların kullanılacağı bir kurallar zinciriyle tanımlanmalı. Tüm bu özellikler, süreçler, ilişkiler ve kurallar bir ontolojide tanımlanabilir. Bu tanımlamalar için ontolojilerde bir takım kavramlar vardır[4] Bunlar;

- Sınıflar (classes); bilgi alanlarındaki genel kavramların tanımlarıdır.
- Örnekler (instances); sınıfların özelliklerini taşıyan öğelerdir.
- İlişkiler (relationships); öğeler arasındaki bağlantıları tanımlar.
- Özellikler (properties); sınıfların ve örneklerin özellikleri tanımlanır.
- Fonksiyonlar (functions); öğelerle ilgili fonksiyonlardır.
- Kısıtlar (constraints); ontolojide tanımlanan nesne ve sınıflarla ilgili kısıtlardır.
- Kurallar (rules); İlişkiler ve özellikler arası kısıtları, alanların alabileceği değerleri (range) vb. tanımları içerir.

3.2. Anlamsal Web Dilleri

Anlamsal web dilleri ontolojilerin ve ontolojilerle web ortamındaki nesnelere (kaynakların) tanımlanmasını sağlar.

3.2.1. XML (Extensible Markup Language)

XML günümüzde kullanılan işaretleme dillerinin büyük bir çoğunluğunun temelini oluşturan genişletilebilir işaretleme dilidir. XML, verilerin transferi, depolanması, sorgulanması ve yönetiminde; veriye içerik değeri katması, ihtiyaç duyduğumuz sistemi yaratabilme esnekliği sunması, dağınık verilerin kümelenmesi, karşılaştırma yapma kolaylığı, farklı veri formatlarını ve dilleri destekleyebiliyor olması ve tüm sistemlerle

çalışabilme özelliğiyle bugün ve gelecekte ihtiyaç duyulan veri standardıdır. Veri öbekleri XML' le bilgiye dönüşür [17].

XML, değişik veri, kavram ve içeriklerin tanımlanması ve temsil edilmesi için uygun bir ortam sunmaktadır. Bu nedenle XML farklı alanlarda uygulama verilerinin tanımlanması ve taşınması için üretici, dil ve platformdan bağımsız stratejik bir araç olarak hızla yaygınlaşmaktadır.

SGML (Standard Generalized Markup Language), ISO (International Standards Organization) tarafından 1986 yılında kabul edilmiş ve onaylanmış bilgi yönetimi standardıdır (ISO 8879:1986). SGML, platform-bağımsız ve uygulama-bağımsız dokümanlar yaratmak ve sağlamak için oluşturulmuştur. Dilbilgisi gibi bir mekanizma kullanılarak, dokümanların yapısını özel tanımlanmış etiketler kullanarak tanımlamamıza yarar. SGML, bir meta dildir. Yani yeni diller üretmeye müsait bir yapıya sahiptir.

Bağımsız bir kuruluş olan W3C organizasyonu tarafından tasarlanan ve herhangi bir kurumun tekelinde bulunmayan XML, Genişletilebilir İşaretleme Dili, SGML 'den türetilmiş, basit ve çok esnek bir işaretleme dilidir. XML başlangıçta büyük ölçekli elektronik yayınlardaki zorluklarla başa çıkmak için tasarlanmıştır. Daha sonraları geniş ölçekli verinin, web ve diğer ortamlar arasında değişiminde önemli rol oynamasıyla kullanımı yaygınlaşmıştır. SGML den türetilmiş işaretleme dilleri, uygulamalar içinde islenecek yada farklı ortamlarda bulunsalar bile uygulamalar arasında dağıtılabilecek verilerin tanımlanmasını ve dağıtımını kolaylaştırmaları bakımından büyük önem taşırlar. XML'in çok kısa zaman içinde veri transferi konusunda evrensel bir protokol haline gelmesinde HTTP (Hypertext Transfer Protocol) kullanmasının da büyük etkisi vardır. Dokümanları internette yayınlamanın en yaygın yolu olan HTML'in (Hyper Text Markup Language) yanında, uygulama geliştiricileri tarafından veri transferi ve islemesi konusunda en çok tercih edilen dil XML'dir.

3.2.2. RDF (Resource Description Framework)

Kaynak Açıklama Çerçevesi (RDF), World Wide Web'deki (www) kaynaklar hakkında bilgiyi ifade etmek için kullanılan bir dildir. Özellikle Web' in bilgi kaynaklarının meta verisi RDF ile ifade edilir. Örneğin bir Web Sayfa' sının, değişiklik tarihi, yazarı, başlığı, bir Web dokümanı hakkında bilginin telif hakkı ve lisans bilgileri yada paylaşılan kaynaklar için kullanılabilirlik zaman çizelgesi RDF ile tanımlanır. Bununla birlikte Web kaynakları kavramını genellersek; RDF, internet üzerinde kimlikleri teşhis edilebilen tüm öğelerin bilgilerini görüntülemek içinde kullanılabilir. Örneğin çevrimiçi alış-veriş sitelerinde ulaşabileceğimiz öğeler hakkında bilgi içerebilir (ürünün özellikleri, fiyatı, stok miktarı vb.)[18].

RDF, bilginin web üzerinde insanlar tarafından görüntülemesinin yanında uygulamalar tarafından işlenebilecek şekilde tutulmasını amaçlamaktadır. RDF bilgiyi ifade etmek için genel geçer bir çerçeve sağlar ve böylece bilgi uygulamalar arasında gezerken anlamını kaybetmez. RDF ortak bir çerçeve sunduğu için uygulama geliştiricileri de ortak RDF çözümleyicileri ve işlem araçları geliştirmek için güç birliği yapıyorlar. Farklı uygulamalar arasında veri değişimi, herhangi bir uygulama için yaratılmış bilginin başka uygulamalar içinde kullanılabilir hale getirilebilmesi anlamına gelmektedir.

RDF Web' deki öğeleri “Uniform Resource Identifier” (URI) adı verilen web tanımlayıcıları tarafından ifade edilmesi ve kaynakların basit özellikler ve özelliklerin değerleri ile açıklanması fikri üzerine bina edilmiştir [19].

RDF web kaynaklarını basit ve anlaşılır bir şekilde cümleler şeklinde tanımlamamızı sağlar. RDF ifadelerinde yer alan nesne, özellik, değer üçlemesi RDF' in temelini oluşturur. Nesne, özellik ve değer üçlüsü RDF de özne (subject), yüklem (predicate) ve nesne (object) olarak ifade edilir.(Şekil 3.5)



Şekil 3.5 RDF Grafiksel Gösterimi

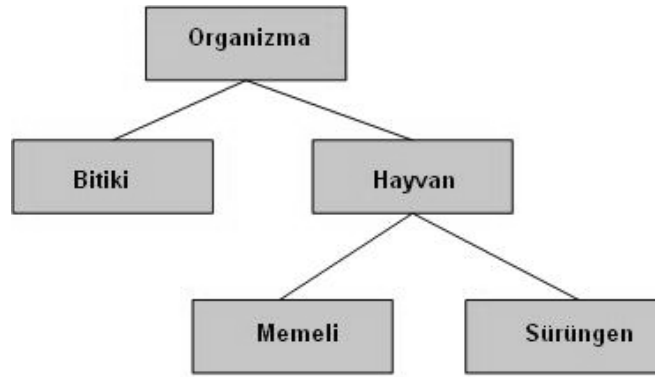
Burada ;

- Özne URIref: <http://www.example.org/index.html>
- Yükleme URIref: <http://purl.org/dc/elements/1.1/creator>
- Nesne URIref: <http://www.example.org/staffid/85740> olarak ayırabiliriz.

RDF ifadeleri özne, yüklem ve nesne üçlüleri olarak da gösterilebilir. “Notation3” olarak adlandırılan üçlü gösterimin anlaşılması kolaydır. Üçlü RDF gösteriminde her bir satır özne, yüklem ve nesne üçlülerini içerir.

3.2.3. RDFS (RDF Schema)

RDFS web ortamındaki kaynaklar, isimlendirilmiş kaynak özellikleri ve değerleri üçlülerini temel alan basit bir veri modeli gösterim yöntemidir. RDFS gösterimi RDF veri modelini genişleten bir tip sistemidir. Bu tip sistemi bir alanda kullanılacak olan sözcük kümesini tanımlar. Bu sözcük kümesi bir alanda kullanılacak olan nesnelere, nesnelere arasındaki alt/üst küme ilişkileri, özellikler, özelliklerin alabileceği değerleri tanımlar (Şekil 3.6)[20].



Şekil 3.6 RDF-S Grafiksel Gösterimi

Bunları tanımlarken RDFS sınıflardan ve özelliklerden veri modeli tanımlama elemanları kullanır. RDFS’ de kullanılan temel sınıflar:

- rdfs:Resource: Resource sınıfı tepe sınıfıdır. Tüm diğer sınıflar bu sınıfın bir örneği olarak yaratılır.
- rdf:Property : Özellikler tanımlanır.
- rdfs:Class: Nesnelere tanımlanır.
- rdfs:Literal: Sabit değerler tanımlanır (ör: tam sayılar, karakter dizileri).

RDFS’de kullanılan temel özellikler:

- rdf:type: Özne bir nesnenin örneğidir (instance) veya tipindedir.
- rdfs:subClassOf: Nesne bir nesnenin alt nesnesidir.
- rdfs:subPropertyOf: Özne bir özelliğin alt özelliğidir.
- rdfs:domain: Özne özelliğinin alanını tanımlar.
- rdfs:range: Özne özelliğinin alabileceği değerler aralığıdır.
- rdfs:label: Öznenin açıklayıcı ismidir.
- rdfs:comment: Özne kaynağı hakkında açıklayıcı tanımlamadır.

3.2.4. OWL (Web Ontology Language)

OWL bir dokümanın sadece insanlar tarafından okunup anlaşılmasının ötesinde doküman içeriğinin uygulamalar tarafından kullanılabilir anlamlı bir veri şeklinde saklanmasını amaçlamaktadır. Kelime hazinemizdeki terimlerin anlamlarını bize açıkça ifade etmesinin yanında, bu terimler arasındaki ilişkileri de tanımlamamızı sağlar. Verinin anlamını ve ilişkilerini ifade etmek adına XML, RDF ve RDF-S gibi işaretleme dillerinden çok daha fazla imkan sunar. Bu durum da OWL' ın, makine tarafından yorumlanabilir bir web içeriği sunma konusunda diğer dillerin çok ötesinde olduğunu göstergesidir[21].

W3C, XML ve RDF'in üzerinde bir web ontoloji dili gereksinimi olduğunu şu sözlerle ifade etmektedir; "Farklı kurumlar arasında veri değişimi veya araması yapan uygulamalar için ontolojiler çok önemlidir. Her ne kadar XML, DTDs ve XML Şemalar daha önceden anlaşılmiş veri yapıları üzerinde veri değişimini mümkün kılarsa da, bu dillerde anlamsallık olmadığından yeni bir XML kelime hazinesi tanımlandığında bunun bilgisayar tarafından algılanıp kullanılması mümkün değildir. Bu sorun basit seviyede anlamsallığı ifade etmemize olanak sağlayan RDF ve RDF Şemalar ile belli bir miktarda çözülmüştür, ancak çok sayıda otonom olarak gelişen ve yönetilen şemalar arasında birlikte çalışılabilirliği sağlamak daha zengin bir anlamsallık gerektirir. Örneğin RDF Şemalarda bir kişinin erkek veya kadın olabileceğini, bir babanın bir kişi ve bir erkek olabileceğini tanımlayabiliriz ancak bir babanın kadın olamayacağını ifade etme olanağımız yoktur." [8]

Genel olarak şunu diyebiliriz ki OWL, XML ve RDF şemalarının sağladığı bütün olanakları sağlar. OWL' ı ayrıcalıklı kılan sınıfları çok daha farklı ve karmaşık yollarla ifade edebilmesidir. Örneğin bitkilerin ve hayvanların ayrık kümeler olduğunu ifade edebiliriz: hiçbir birey aynı anda hem bitki hem de hayvan olamayacağını yada tuzlu su balığının, balıklar ve deniz canlılarının kesişim kümesi olduğunu ifade edilebilir.

OWL birey (individual), özellik (property) ve sınıf (class) bileşenlerinden meydana gelmektedir.

- Birey domain içerisinde ilgilendiğimiz nesnelere dir. Bireyler aynı zaman da bir sınıfın örneği (instance) olarak da ifade edilmektedir.
- Özellikler bireyler arasında ikili ilişkiler kurmak için kullanılır. Örneğin A bireyi B bireyini kapsar yada B bireyi C bireyinden farklıdır gibi.
- Sınıflar bireyleri içeren kümeler olarak yorumlanabilir. Bir sınıfa ait özellikleri taşıyan tüm bireyler o sınıfın bir örneğidir. Sınıfları alt sınıf üst sınıf şeklinde hiyerarşik bir yapıda ilişkilendirebiliriz. Alt sınıf kendisinin üst sınıfının özelleşmiş bir halidir. Bir alt sınıfa ait birey aynı zamanda üst sınıfının da bir örneğidir.

OWL bir Web Ontoloji dili ihtiyacına karşılık olarak tasarlandı ve W3C tarafından anlamsal web ile ilintili olarak önerilmektedir [21]. Konuyu maddeleyecek olursak:

- XML dokümanların yapısını tanımlayabilmemiz için bize bir söz dizimi sağlar ancak bu söz dizimi kısıtları veya öğelerin anlamlarını ifade edebilecek bir anlamsallıktan yoksundur.
- XML şemalar XML doküman yapılarına kısıtlar getirir ve ayrıca XML veri tiplerinin tanımlanmasını sağlar.
- RDF nesnelere (kaynaklar) ve onlar arasındaki ilişkileri tanımlayabilmemiz için bize basit seviyede bir anlamsallık kazandıran veri modelidir. Bu veri modeli XML söz dizilimi ile gösterilir.
- RDF Şema RDF kaynaklarının özelliklerini ve sınıflarını belli bir hiyerarşide tanımlamamızı sağlayan dildir.
- OWL özellikleri ve sınıfları tanımlayabileceğimiz daha geniş bir kelime hazinesi sağlamaktadır. ; RDFS ile bireyler için hiyerarşik yapıda bir özellikler zinciri tanımlarken OWL ile geçişli, simetrik veya fonksiyonel özellikler tanımlayabiliriz

OWL OWL Lite, OWL DL ve OWL Full olmak üzere üç alt dil sağlar:

- OWL Lite kullanıcılar için temel olarak gereken sınıflandırma hiyerarşisi ve basit kısıtlar sağlar. Örneğin önem derecesi (cardinality) kısıtı için sadece 0-1 değerlerinin verilebilmesine olanak sağlar.
- OWL DL üst seviyede anlamsallık isteyen kullanıcıların taleplerini karşılar. OWL DL OWL dilinin tüm yapılarını destekler. Fakat OWL dilinin tüm kısıtlarını desteklemez. Örneğin bir sınıf bir çok sınıfın alt sınıfı olabilir ancak başka bir sınıfın örneği olamaz. Bu özelliği ile OWL DL bütün durumlar için hesaplana bilirliği garanti eder.
- OWL Full tam anlamsallığın yanında sözdizimi için özgürlük isteyen kullanıcılar içindir. Ancak OWL Full her durum için hesaplana bilirliği garanti etmez. Bunun yanında sözdizimi serbestliği sayesinde OWL DL ile tanımlayamayacağımız kısıtlar OWL Full ile tanımlanabilir. Örneğin OWL Full’ de bir sınıf aynı zamanda bir nesnelere kümesi olarak davranabilir yada bir nesne sınıf olarak kullanılabilir.

3.3. Anlamsal Web Servisleri

WSDL [10] ve SOAP [22] gibi genel kabul görmüş standartlara sahip olması web servislere dinamik olarak ulaşabilmemizi ve çağırabilmemizi mümkün kılar.

Kullanılacak olan servisin anlamı(semantiği) bilindiği zaman, WSDL açıklamasına ve bununla birlikte servisin fonksiyon ve özelliklerine bir program tarafından erişilebilir. Web servisin bütün işlevselliğiyle kullanıma sunulması için servise ait anlamsal bilginin çok iyi verilmiş olması gerekmektedir. Bu şekilde çok iyi tanımlanmış bir servis insanlar tarafından kolaylıkla kullanılabilir. Bundan daha da önemli olan nokta ise, web servisleri otomatik olarak keşfedip işleyen yazılım ajanlarının kaliteli anlamsal bilgilerle önünün açılmasıdır.

3.3.1. OWL-S

OWL-S, web servis sağlayıcılarına kendi web servislerinin özelliklerini ve yapabileceklerini, karşılıklıya meydan vermeyecek ve bilgisayar tarafından

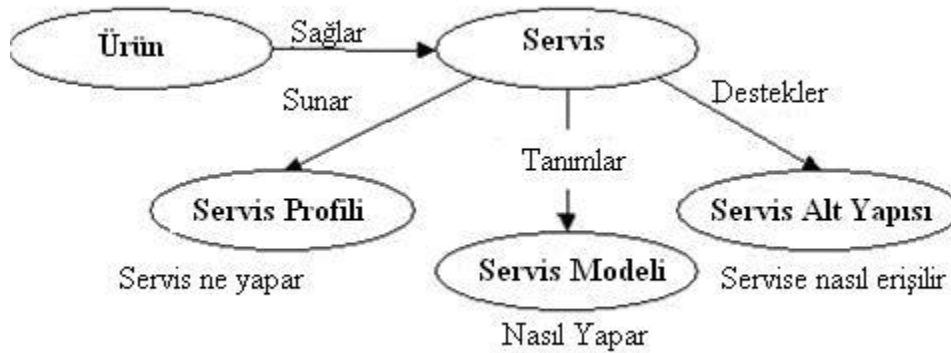
yorumlanabilir şekilde tanıtılabilecekleri temel bir işaretleme dili sağlayan OWL web servis ontolojisidir. OWL-S mimarisinin amacı web servislerin keşfedilmelerinin, çalışmalarının ve izlenmesinin otomatikleştirilmesini kolaylaştırmaktır[23].

OWL-S web servisler iki kategoriye ayrılır:

- “Temel”: Web tarafından erişilebilir olan tek bir bilgisayar programı çağırılır. Kullanıcı ve servis arasında basit bir cevaptan öte bir etkileşim olmaz.
- “Kompleks(Karışık)”: Birçok “temel” servisten oluşur. Kullanıcının seçim yapabilmesi ve istediği şarttaki bilgileri alabilmeleri için kullanıcı ile servis arasında sık sık etkileşim ve bağlantı gerektirir.

OWL-S, aşağıdaki işlevleri kolaylaştırmak için iki kategorideki servisleri de desteklemektedir.

- Otomatik web servis keşfetme; istenen özellik ve yapıdaki web servislerin keşfedilmesidir.
- Otomatik web servis yürütme; web servisin bir bilgisayar programı ya da yazılım ajanı tarafından otomatik çalıştırılmasıdır.
- Otomatik web servis bileşimi; web servislerin belirli bir amaç doğrultusunda otomatik olarak seçilmesi, çalıştırılması ve kendi aralarında koordineli olarak çalışmasıdır.
- Otomatik web servis çalışma monitörü; kendi başlarına çalışan web servisler ya da bir amaç doğrultusunda beraber çalışan web servisler belirli bir zaman sonra çalışmalarını bitirmelidirler. Kullanıcılar isterlerse bu belirli zaman aralığında servislerin durumunu öğrenebilir, isterlerse web servislerde değişiklik yapabilirler.



Şekil 3.7 OWL-S Üst Seviye Servis Ontolojisi

OWL-S, servis tanımlama için bir üst ontoloji sağlar(Şekil 3.7). Bu ontolojideki en üst seviye sınıf “Service” sınıfıdır. “Service” sınıfının aşağıda belirtilen 3 özelliği vardır:

- “sunar”; bu özelliğin alanı ServiceProfile sınıfıdır. Service sınıfı, kullanıcılara nasıl bir servis sağladığını belirtmek ve kullanıcılardan ne tür bilgileri alması gerektiğini bildirmek üzere bu sınıfı sunmaktadır.
- “tanıtılır”; bu özelliğin alanı ServiceModel sınıfıdır. Service sınıfı, ServiceModel sınıfı tarafından nasıl çalıştığını belirtmek üzere tanıtılır.
- “destekler”; bu özelliğin alanı ServiceGrounding sınıfıdır. Service sınıfı, nasıl kullanıldığını desteklemek amacı ile ServiceGrounding sınıfını kullanır.

BÖLÜM 4. UYGULAMA

4.1. Kullanılan Araçlar

Tezin bu bölümünde örnek uygulamanın geliştirildiği ve idamesinin yapıldığı yazılım araçları tanıtılacaktır.

4.1.1. Protege

Protege ontoloji modelleme ve bilgi çıkarımı (knowledge acquisition) için kullanılan, Stanford Üniversitesi Medical Informatics'te geliştirilen açık kaynaklı bir platformdur. Sayısı 20000'i bulan geniş bir kullanıcı kitlesine sahiptir

Karışık ve çabuk hata yapmaya sebep olan sözdizimine sahip RDF'i temel alan OWL ile ontoloji geliştirmek zordur. İnsan küme bilgisinin formal modelleri olan ontolojilerin geliştirilmesini kolaylaştıran ve hata yapılma olasılığını azaltan araçlar ontoloji geliştirilmesinde büyük kolaylıklar sağlayacaktır. Bu araçlar ayrıca kolay anlaşılabilen ve farklı uygulamalar için özelleştirilebilecek açık kodlu araçlar olmalıdırlar.

Birçok modelleme aracında olduğu gibi Protege mimarisi model ve görünüş olarak iki parçaya bölünmüştür. Model ontolojilerin içsel gösterimleri ve bilgi tabanlarıyla ilgili mekanizma, görünüş ise bu modellerin görüntülenmesi ve işlenmesine olanak verecek bir arayüz sağlar.

Protege sınıfların (classes), özelliklerin (properties, slots), özellik karakteristiklerinin (property characteristics, facets ve constraints) ve örneklerin temsil edilmesine izin verir. Sağladığı Java API'si sayesinde ontolojilerin işlenmesi ve sorgulanmasına olanak sağlar.

Protege'in görünüş kısmında ontoloji tasarımcılarının kolay şekilde sınıflar yaratabilmeleri, bu sınıflara özellik atamaları ve bu özellikler üzerinde kısıtlama yapabilmeleri sağlanır.

Protege CLIPS, RDF, XML, UML ve ilişkisel veritabanlarında ifade edilmiş ontolojilerin işlenmesini (load and manipulate) desteklemektedir. Bunun dışında Protege olay mekanizmaları (event mechanisms), geri dönme (undo) yetenekleri ve eklenti (plugin) mekanizmaları gibi temel özellikleri de sağlamaktadır. OWL Plugin bir Protege eklentisi olarak Protege'in sağladığı istemci/sunucu çok kullanıcı modunu birçok kullanıcının aynı anda aynı ontoloji üzerinde çalışabilmelerine izin vermek için kullanılmaktadır. Ek olarak Protege sürekli büyüyen ve gelişime destek veren bir kullanıcı topluluğuna sahiptir.

4.1.2. Jena

Jena anlamsal web uygulamaları geliştirmek için kullanılan bir java çerçevesidir. RDF, RDFS, OWL, SPARQL için programlama ortamı sağlar. Ayrıca kural tabanlı bir çıkarsama motoru içerir[2].

Jena çerçevesi;

- RDF uygulama program ara yüzü içerir,
- RDF'leri RDF/XML, N3 ve N-Triples olarak okumamıza ve yazmamıza olanak sağlar,
- OWL uygulama program arayüzü içerir,
- SPARQL sorgulama moturu içerir.

Jena çatısı, ontolojileri yorumlar ve ontolojilerin özünü oluşturan çizge yapısını oluşturularak, bu yapının bir model içerisinde tutulmasını sağlar. Bu modele Jena model denmektedir. Jena çatısının anlamsal web portal içerisinde tercih edilmesinin sebepleri ücretsiz olarak kullanıma açık olması ve Java dili desteğidir. Jena çatısı, anlamsal web portal mimarisi içerisinde, ontoloji nesnelere oluşturulması ve bu nesnelere ilgili

işlemlerin tamamlanmasında kullanılmaktadır. Jena çatısını kullanan modüller sayesinde, ontolojilerin ayrıştırılması ve işlenmesi, oluşan ontoloji modellerinin ve nesnelere diğer modüllerde kullanılabilmesi mümkün olmaktadır.

4.1.3. NetBeans

Netbeans, çok geniş bir kullanıcı tabanı, büyüyen topluluğu, dünya çapında yüze yaklaşan ortağı olan başarılı bir açık kaynak kod projesidir. Sun Microsystems, NetBeans açık kaynak kodlu projesini Haziran 2000 tarihinde başlatmış olup halen projelerin ana sponsorudur. Günümüzde iki ürünü bulunmaktadır : NetBeans IDE'si ve NetBeans Platformu.

NetBeans IDE'si programcıların yazma, derleme, hata bulma ve yüklemelerini sağlayan bir araç olan geliştirme ortamıdır. Java ile yazılmış olmasına rağmen herhangi bir programlama dilini destekleyebilir. NetBeans IDE'sini genişletmek için çok sayıda modül bulunmaktadır. NetBeans IDE, nasıl kullanıldığına ilişkin hiçbir sınırlama bulunmayan ücretsiz bir üründür.

Ayrıca halihazırda bulunan NetBeans Platformu, modüler ve genişletilebilir yapısıyla büyük masaüstü yazılımları üretmede kullanılır. ISV ortakları platforma kolayca entegre edilebilen pluginler sağlarlar. Bunlar kendi araç ve çözümlerini gerçekleştirmede de kullanılabilir.

Her iki ürün de açık kodlu olup kar amaçlı olsun olmasın kullanılabilir. Kaynak kod Common Development and Distribution License (CDDL) altındadır.

4.2. Uygulama Tasarımı

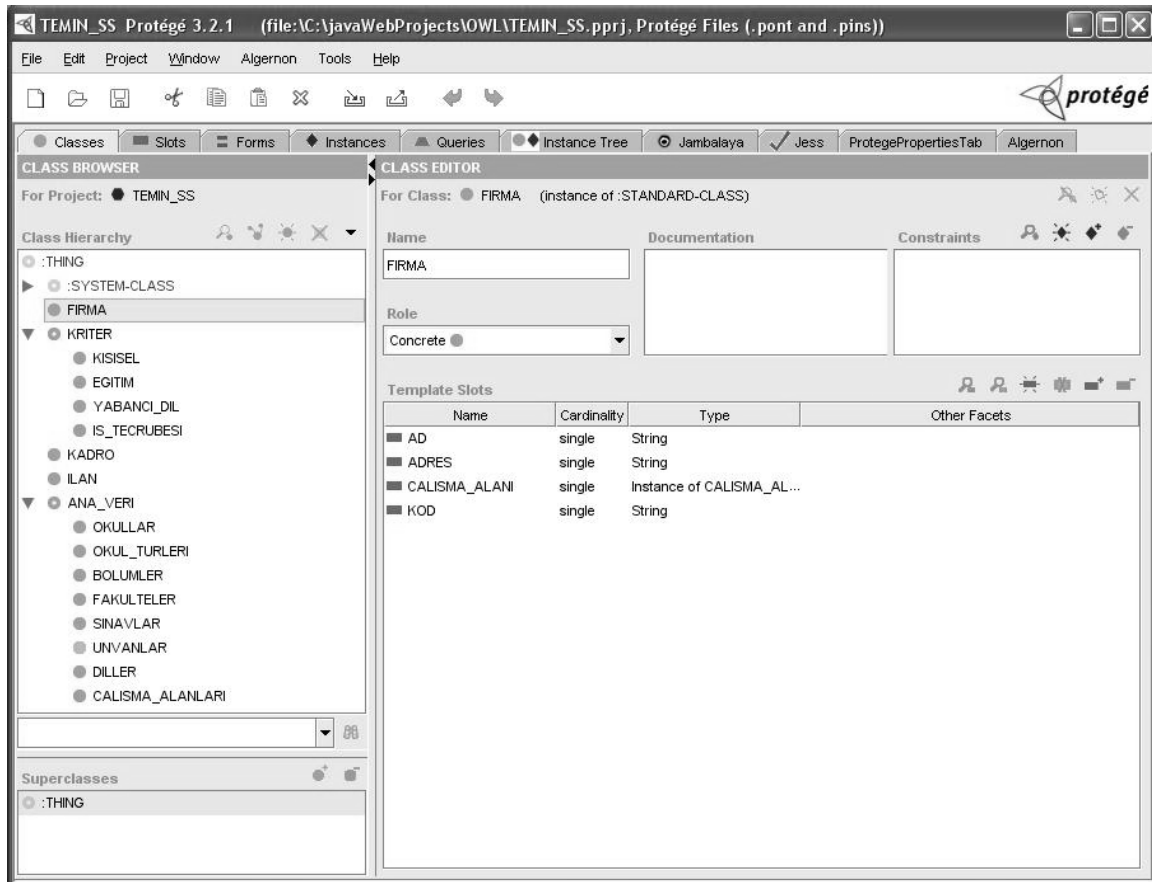
Gerçekleştireceğimiz uygulamada bir firmanın yayınlayacağı bir iş ilanına başvuran adayın uygun olup olmadığını çevrimiçi olarak tespit edeceğiz. Adayın uygun olduğu durumda bilgileri yine çevrimiçi olarak ilanı veren firmaya gönderilecek. Uygun

olmadığı durumda ise adayın uygun olmayan bilgisi kendisine gösterilerek başvuru işlemi olumsuz olarak sonlandırılacaktır.

Firma kriterlerini tasarlanan temin ontolojisine göre belirleyecek. Aday başvurusunu yaptığında devlet kurumlarının (ÖSYM, MEB, MERNİS vb.) sağladığı web servisler aracılığıyla bilgileri sorgulanarak firmanın belirlediği kriterlere uygunluğu tespit edilecektir.

4.2.1. Kullanılan ontoloji

Uygulamada firmaların ilan verecekleri kadroların kriterleri belirlemeleri için “Protege”[24] aracı kullanılarak bir ontoloji tasarlanmıştır.(Şekil 4.1.)

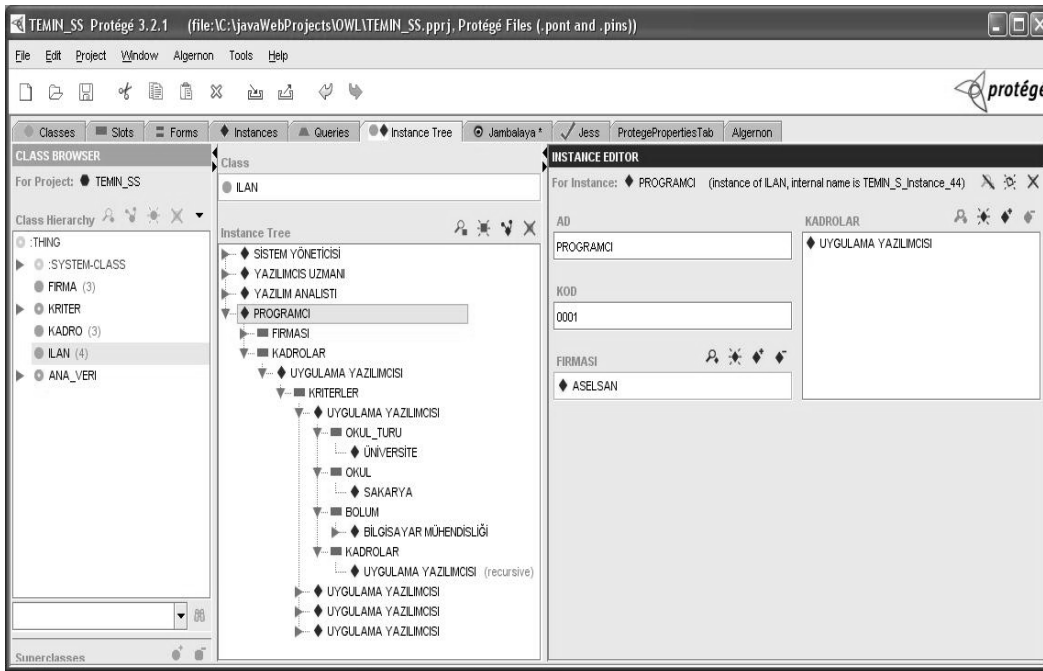


Şekil 4.1 Personel Alım Ontolojisi

Tasarlanan ontoloji ařađıda belirtilen sınıf ve özelliklerden oluşmaktadır.

- FIRMA: Başvuru sistemine kayıtlı firmaların tanımlandığı sınıftır. Bu sınıfa bađlı olarak ad, firma kodu, adres ve çalışma alanı özellikleri bulunmaktadır.
- KADRO: Başvuru yapılacak kadroların tanımlandığı sınıftır. Kadro sınıfına kadro adı ve kriter sınıfından oluşacak örnekler, özellik olarak bađlanmıştır.
- İLAN: sisteme girilecek olan ilanlar bu sınıf aracılığıyla tanımlanır. İlan sınıfına ilan başlama tarihi, ilan bitiş tarihi, ilanı veren firmanın tanımlanması için firma sınıfından bir örnek ve ilan verilen kadroların belirtilmesi için kadro sınıfının örnekleri özellik olarak bađlanmıştır.
- KRITER: İlana muhatap kadroların özellikleri bu sınıf aracılığıyla tanımlanır. Dört tane alt sınıf içerir (istenirse yeni kriter sınıfları eklenebilir). Bunlar EGITIM, KISISEL, YABANCI_DIL ve IS_TECRUBESI sınıflarıdır.
- EGITIM: Kadronun eğitimle ilgili kriterlerinin belirlendiđi sınıftır. Alınmış olan eğitimlerin okul türünü (ilk öğretim, lise, üniversite vb.), okul kodunu, bölüm kodunu, mezuniyet notunu, eğitimin başlama ve bitiş tarihleri bu EGITIM sınıfının özellikleridir.
- KISISEL: Kadronun beklediđi kişisel bilgilerle ilgili kriterler KISISEL sınıfıyla tanımlanır. Bu sınıfta doğum tarihi ve cinsiyet bilgileri bulunur.
- YABANCI_DIL: Beklenen yabancı dil bilgisi bu sınıfın özellikleriyle tanımlanır. Dil kodu, sınav kodu (ÜDS, KPDS vb.), sınav tarihi ve sınav notu özelliklerini içerir.
- IS_TECRUBESI: Tanımlanan kadro için beklenen iş tecrübesinin belirtildiđi sınıftır. Firma kodu, meslek kodu, iş unvanı, başlama tarihi ve bitiş tarihi özelliklerini içerir.
- ANA_VERI: Kriterlerin özelliklerinde kullanılan değerlerin ana verilerinin tanımlandığı sınıftır.

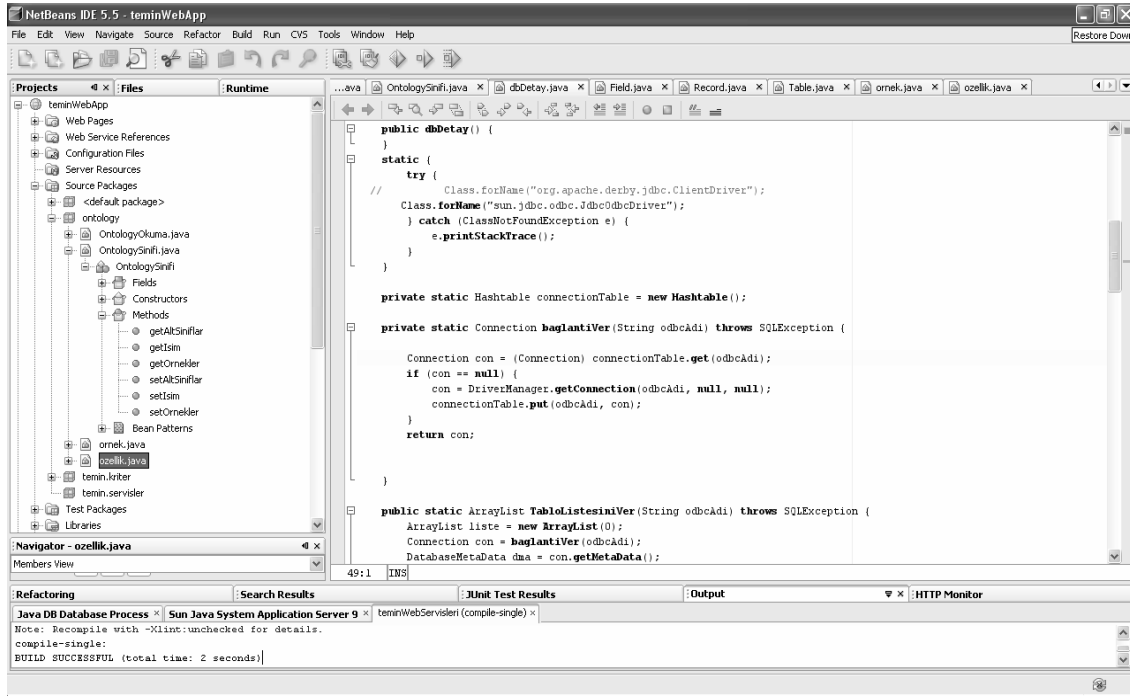
Yukarıda belirtilen sınıflardan üretilmiş örnek ağacı Şekil 4.2'de gösterilmektedir.



Şekil 4.2 Personel Alım Ontolojisi Örnek Ağacı

4.2.2. Sınıflar ve metotları

Tasarlanan uygulamada veri tabanı tablolarını okumak ve ontolojiyi çözümlmek için java platformu kullanıldı. Sınıfları ve web ara yüzlerini geliştirmek için “NetBeans IDE 5.5” aracı kullanıldı (Şekil 4.3) [25].



Şekil 4.3 Uygulama geliştirme ortamı NetBeans IDE 5.5

Veri tabanının tablo yapılarını ve kayıtlarını okumak için kullanılan sınıflar:

- Field sınıfı: Tablo alanlarının adının, tipinin ve uzunluğunun okunmasında kullanılan sınıftır.
- Record sınıfı: Field sınıfının ve bu sınıfa bağlı olarak tabloda ki kaydın okunması için kullanılan sınıftır
- Table sınıfı: İçerisinde tablo adını, field sınıfını ve record sınıfını bulundurur.
- dbDetay sınıfı: iki parametre alır. Bunlardan biri veri kaynağı diğeri sorgu cümlesidir. Verilerin parametreleri kullanarak veri kaynağı altındaki herhangi bir tabloyu verilen sorgu cümlesine göre sorgulayarak sonuç getiri. Dönen sonuçta sorgulanan tablonun veri yapısı ve kayıtlar bulunur.

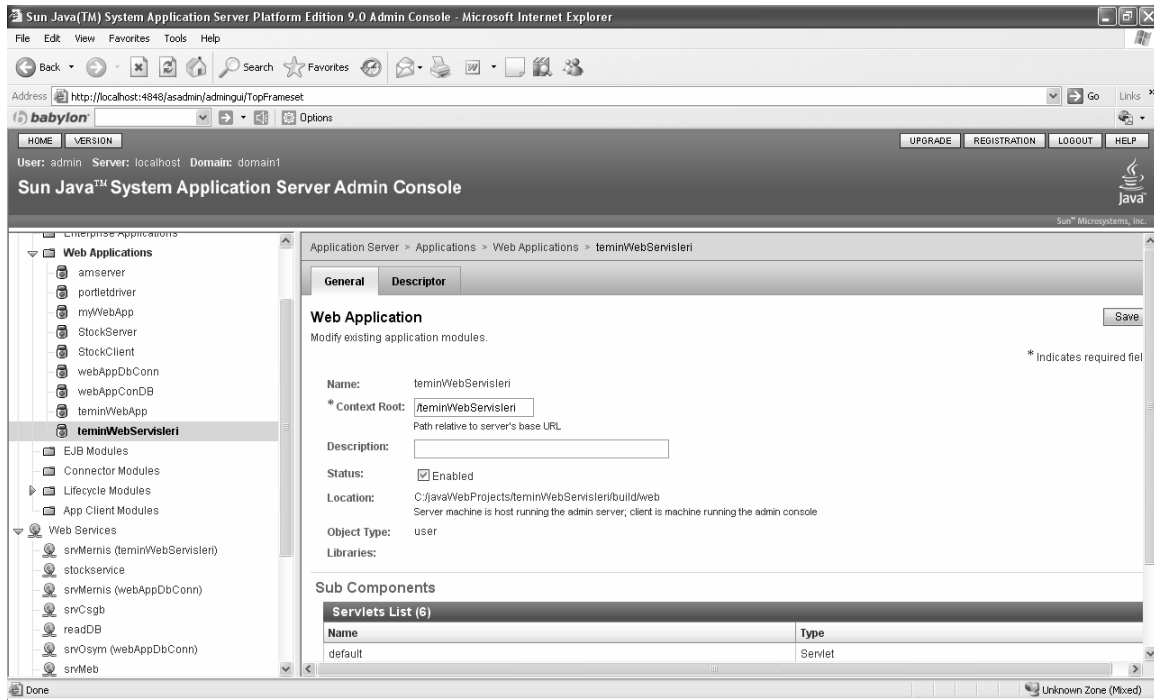
Ontolojiyi, örneklerini ve özelliklerini okumak için kullanılan sınıflar:

- Ornek sınıfı: Ontoloji içerisinde bulunan örnekleri ve bunlara ait özellikleri (Ozellik sınıfı) içerir.
- Ozellik sınıfı: bir örneğin her bir özellikle olan bağlantı ifadesi özne, yüklem ve nesne olarak tutulur.
- OntologySinifi sınıfı: Ontolojideki sınıfları ve sınıflara bağlı olan alt sınıfları dizi olarak ifade etmek için kullanılır.
- OntologyOkuma sınıfı: Dosya adresi verilen ontolojiyi OntologySinifi ve Ornek sınıflarını kullanarak ayrıştırır. Sonuç olarak içerisinde diğer sınıfları bulunduran bir dizi döndürür.

4.2.3. Web servisleri

Uygulamada J2EE[26] platformu üzerinde “NetBeans IDE 5.5” aracı kullanılarak tasarlanan ve “Sun Java System Application Server Platform Edition 9.0_01” [27] sunucusunda yayınlanan web servisler kullanıldı (Şekil 4.4). Farklı kurumların sağladığını var saydığımız bu web servisler:

- srvMernis: Merkezi nüfus işleri idaresinin sağladığı varsayılan web servisidir. Giriş parametresi olarak T.C. kimlik numarası alır ve sonuç olarak adayın kişisel bilgilerini döndürmektedir.
- srvOsym: Öğrenci Seçme ve Yerleştirme Merkezi’ nin sağladığı varsayılan web servisidir. Servis giriş parametresi olarak T.C.kimlik numarası alıyor. Sonuç olarak adayın dil sınav sonuçlarını döndürmektedir.
- srvMeb: Milli Eğitim Bakanlığının sağladığı varsayılan web servisidir. Verilen T.C. kimlik numarası karşılığında adayın tüm resmi eğitim bilgilerini vermektedir.
- srvCsgb: Çalışma ve Sosyal Güvenlik Bakanlığının sağladığını varsaydığımız web servisidir. T.C. kimlik numarası girilen adayın hangi sosyal güvenlik kurumuna bağlı olarak, hangi kuruluştaki, hangi tarihler arasında çalıştığının bilgisini getirmektedir.



Şekil 4.4 Kullanılan Uygulama Sunucusu Sun Java System Application Server Platform Edition 9.0_01

4.2.4. Web sayfaları

Uygulamamızda biri firmalar diğeri adaylar tarafından kullanılmak üzere iki web sayfası bulunmaktadır. Bunlar:

- ilan.jsp: Firmaların ilan girişini yaptığı web sayfasıdır. Bu sayfada firmalar alım yapacakları kadrolar için belirledikleri kriterleri tanımlarlar.
- is_basvuru.jsp: Bu sayfa adayların firma ve firmanın ilan olarak girdiği pozisyonu seçerek T.C. kimlik numaraları ile başvuru yaptığı sayfadır (Şekil 4.5).



Şekil 4.5 İş Başvuru Sayfası

4.3. Uygulamanın Gerçekleştirimi

Uygulama temel olarak üç aşamadan oluşmaktadır:

- Ontolojide kullanılan (Firmalar, okullar, bölümler vb.) ana verilerin oluşturulması,
- İlan verecek firmanın ontolojide belirtilen kriterler ışığında ilanlarını yayınlaması,
- Adayın T.C. Kimlik numarasını kullanarak seçtiği ilana başvurması ve başvurunun sonuçlanması.

İlk aşamada ontolojide bulunan ana verilerle ilgili sınıflar için örnekler oluşturulacaktır. Bu sınıflar kriterlerde kullanılan sınıflar için bir bilgi kümesi olarak düşünülebilir.

Örneğin firmalar sınıfında firmaların kodu, adı, çalışma alanı tanımlanacaktır. Daha sonra ilan sınıfından oluşturulan örnekler firma kodu alanı üzerinde firma sınıfı örneklerine ulaşacaklardır. Bu ilk aşama sisteme yeni bir firma eklenmesiyle tetiklenecektir.

İkinci aşamada sistemde tanımlı olan firmalar ilan sınıfından bir örnek yaratacaklardır. Bu örneğe bağlı olarak kadro sınıfından ve kriter sınıfından da örnekler yaratacaklar. Örneğin ilan sahibi firma, kadro sınıfında tanımladığı “Uygulama Programcısı” örneği için kriter sınıfının alt sınıflarında örnekler yaratarak; “Üniversitelerin Bilgisayar Mühendisliği Bölümü Mezunu”, “30 Yaşını Aşmamış, Erkek”, “Programcılık Alanında 3 Yıl Tecrübeli”, “Son 2 Yıl İçerisinde, KPDS’den 60 ve Üzeri Puan Alan Adaylar” şeklinde bir kadro tanılaması yapabilir.

Son kısımda ise adaylar başvuru sayfasına bağlanarak firmaların yayınladığı iş ilanlarını ve aranan kriterlerini görüntüleyecekler. Kendilerine uygun buldukları kadroyu seçerek sadece T.C. kimlik numaralarını sisteme girerek başvurularını gerçekleştirecekler. Başvuru yapan adayın bilgileri T.C. kimlik numarası parametresiyle kurumların web servileri çağırılarak sorgulanacak ve toplanan bilgiler ontoloji vasıtasıyla tanımlanan kriterlerle karşılaştırılacak. Aday pozisyona uygunsa başvuru olumlu olarak sonuçlanacak. Adayın bilgileri uygun değilse uygun olmayan bilgisiyle birlikte bir mesaj gösterilecek ve başvuru olumsuz olarak sonuçlanacak. Örneğin yukarıda belirttiğimiz kriterler için başvuru yapan adayın KPDS notu yoksa; “Son 3 yıl içerisinde KPDS’den 60 ve üzeri puanınız bulunmadığı için başvurunuz başarısız olarak tamamlanmıştır.” şeklinde bir açıklamayla başvuru sonuçlanacaktır.

BÖLÜM 5. SONUÇ VE ÖNERİLER

Anlamsal web, internetin hacminin hızlı bir şekilde büyümesiyle gün geçtikçe daha çok duyacağımız bir kavram olacaktır. Diğer alanlarda olduğu gibi insan kaynakları konusunda anlamsal webin kullanımı yaygınlaşacaktır.

Yapılan çalışmada belirli kısıtlarla hazırlanan personel alım ontolojisi ile personelin pozisyona uygun olup olmadığı belirlendi. Personel adayları ile ilgili bilgiler, güvenilirliği yüksek veri kaynaklarından web servisleri ile alındı. Böylece başvuru sırasında kaybedilen zamandan kazanç sağlandı, kriterleri uygun olmayan adayın başvurusunu firmanın insan kaynakları personeli bilgileri incelemeden sonuçlandı ve ilan sahibine başvurusu kabul edilen personelle ilgili tüm bilgiler doğru kaynaktan sağlanmış oldu.

Daha geniş çalışmalarla yapılacak kapsamlı bir ontolojiyle daha fazla sektöre hizmet sunabilecek uygulamalar geliştirilebilir. Ayrıca tanımlanan kriterlere bir takım değer aralıkları vererek, hangi pozisyonun hangi adaya uygun olduğu bilgisinin ötesinde, ne kadar uygun olduğu bilgisi oluşturulabilir. Böylece adaya sadece uygun pozisyonları değil en uygun pozisyonu, ilan sahibine de uygun adayları değil en uygun adayı gösterme imkanına sahip olabiliriz.

KAYNAKLAR

- [1] Malgorzata Mochol, Radoslaw Oldakowski, Ralf Heese, Ontology based Recruitment Process
- [2] JENA: A Resource Description Framework (RDF)Application Programming Interface (API), June 2004, [http://www.hpl.hp.com/semweb/do/tutorial/RDF API/](http://www.hpl.hp.com/semweb/do/tutorial/RDF_API/), 2008
- [3] Malgorzata Mochol, Holger Wache, Lyndon Nixon, Improving the recruitment process through ontology-based querying
- [4] The Semantic Web, 2003, Michael C. Daconta, Leo J. Obrst, and Kevin T. Smith
- [5] Building Web Services with Java™: Making Sense of XML, SOAP, WSDL, and UDDI, December 2001, Sams Publishing
- [6] Developing Java™ Web Services, 2003 Ramesh Nagappan, Robert Skoczylas, Rima Patel Sriganesh
- [7] SOA Registry ,<http://www.sun.com/products/soa/registry/>, 2008
- [8] W3C: World Wide Web Consortium, June 2003, <http://www.w3.org>, 2008
- [9] WSDL Kavramsal Modeli,www.xml.org, 2008
- [10] Web Service Description Language (WSDL), June 2003, <http://www.w3.org/TR/wsdl>, 2008
- [11] UDDI: Universal Description, Discovery and Integration, June 2003, <http://www.uddi.org>, 2008
- [12] ebXML Registry Information Model v2.1, June 2003, <http://www.ebxml.org/specs/ebRIM.pdf>, 2008
- [13] ebXML, June 2003, <http://www.ebxml.org>, 2008

- [14] ebXML Registry Services Specification v2.1, June 2003, <http://www.ebxml.org/specs/ebiRS.pdf>, 2008
- [15] T.Berners-Lee, J.Hendler, and O.Lassila. The Semantic Web, Scientific American, May 2001
- [16] Semantic Web, June 2003, <http://www.w3.org/2001/SW/>, 2008
- [17] Extensible Markup Language (XML), June 2003, <http://www.w3.org/XML/>, 2008
- [18] RDF, June 2003, <http://www.w3.org/RDF/>, 2008
- [19] RDF Syntax: Resource Description Framework Model and Syntax Specification, W3C Recommendation, June 2003, <http://www.w3.org/TR/REC-rdf-syntax>, 2008
- [20] RDF Schema: Resource Description Framework Schema Specification, W3C Proposed Recommendation, June 2003, <http://www.w3.org/TR/PR-rdf-schema>, 2008
- [21] OWL Web Ontology Language Overview, W3C Proposed Recommendation 15 December 2003, <http://www.w3.org/TR/2003/PR-owl-features-20031215>, 2008
- [22] SOAP: Simple Object Access Protocol, June 2003, <http://www.w3.org/TR/SOAP>
- [23] OWL-S 1.0 Release, January 2003, <http://www.daml.org/services/owl-s/1.0>, 2008
- [24] Protégé Tool, June 2004, <http://protege.stanford.edu>, 2008
- [25] Java Technology, June 2003, <http://java.sun.com/>, 2008
- [26] Java 2 Platform, Enterprise Edition, June 2004, <http://java.sun.com/j2ee/>, 2008
- [27] Java Servlet Technology, June 2004, <http://java.sun.com/products/servlet/>, 2008
- [28] DAML Ontology Library, June 2003, <http://www.daml.org/ontologies>, 2008

- [29] DAML Services Coalition (A.Ankolekar,M.Burstein,J.Hobbs,O.Lassila, D.Martin, S.McIlraith, S.Narayanan, M.Paolucci, T.Payne, K.Syara, H.Zeng),DAML-S: Semantic Markup for Web Services, In Proceeding of the International Semantic Web Working Symposium (SWWS), July 2001.
- [30] Hypertext Transfer Protocol (HTTP), June 2003, <http://www.w3.org/Protocols/>, 2008
- [31] IBM UDDI registry, July 2003, <http://www3.ibm.com/services/uddi/>, 2008
- [32] IBM's tutorial on Web Services, July 2003, <http://www.ibm.com/software/solutions/webservices>, 2008
- [33] IHE IT Infrastructure Technical Framework, vol. 1 (ITI TF-1) , 2008
- [34] J2EE Tutorial, January 2003, <http://java.sun.com/j2ee/1.4/docs/tutorial/doc/>, 2008
- [35] Java API for XML-Based RPC, June 2004, <http://java.sun.com/xml/jaxrpc/index.jsp>, 2008
- [36] JXTA Bridge, July 2003, <http://www.peerfear.org/jxta-bridge/>, 2008
- [37] MAFRA Toolkit, March 2004, <http://sourceforge.net/projects/mafra-toolkit/>, 2008
- [38] Michel Bourse, Mounira Harzallah, Michel Leclère, Francky Trichet, CommOnCV: competency@ontology.cv, 2008
- [39] OASIS ebXML Registry Reference Implementation Project (ebxmlrr), October 2003, <http://ebxmlrr.sourceforge.net>, 2008
- [40] Oracle Corporation, November 2004, <http://www.oracle.com>, 2008
- [41] Project JXTA 2.0: Java Programmers Guide, March 2004, http://www.jxta.org/docs/JxtaProg-Guide_v2.pdf, 2008
- [42] Project JXTA, June 2003, <http://www.jxta.org>, 2008
- [43] Retired State Employees, Project Announcements; November
- [44] Retired State Employees, Turkey, October 2004 ; <http://www.emekli.gov.tr>, 2008
- [45] SunOne, June 2003, <http://www.sun.com/software/sunone/faq.html#2>, 2008

- [46] UN/CEFACT, July 2004, <http://www.unece.org/cefact>, 2008
- [47] Version 3 Standard: Clinical Document Architecture Framework, Release 1.0, July 2004, <http://www.hl7.org/>, 2008
- [48] Web Services, Axis Project; July 2004, <http://ws.apache.org/axis/>, 2008

ÖZGEÇMİŞ

Atilla ÖZPALA 02.08.1981 de Almanya'nın Frankfurt şehrinde doğdu. İlk, orta ve lise eğitimini Bursa'da tamamladı. 1999 yılında Bursa Nilüfer Milli Piyango Anadolu Lisesinden mezun oldu. 1999 yılında başladığı Sakaya Üniversitesi Bilgisayar Mühendisliği bölümünü 2003 yılında bitirdi. 2004 – 2005 yılları arasında Belsis Yazılım Tic. Ltd. Şti.nde yazılım mühendisi olarak çalıştı. Şu anda, 2005 yılında çalışmaya başladığı Hv.K.K.lığı Bilgi Sistemleri Merkezinde, Entegrasyon Yöneticisi olarak görev yapmaktadır.