

Akıllı Anlamsal Ağ Servisi Arayıcısı

Duygu ÇELİK
Computer Engineering Department
Eastern Mediterranean University
Magusa TRNC, Mersin-10
Email: duygu.celik@emu.edu.tr

Atilla ELÇİ
Computer Engineering Department
Eastern Mediterranean University
Magusa TRNC, Mersin-10
Email: atilla.elci@emu.edu.tr

Özetçe

Bu makale, ağ kullanıcılarının isteğine bağlı olarak, 'Semantic Web Services' yani diğer bir adıyla Anlamsal Ağ Servislerinin (AAS) keşfedilmesini sağlayan, akıllı bir mekanizmadan bahsetmektedir. Ağ servislerinin sayıca artışı ve anlamsal tabanda bir arama yapamaması yüzünden kullanıcının isteğine uygun ağ servislerinin, UDDI deposundan elde edilmesi oldukça zordur. Önerilen sistemin omurgası, iki farklı araştırma konusundan yararlanılarak oluşturulmuştur: Akıllı Ağ Sorgu Motoru ve Uyum Algoritması. Farklı alanlar için önerilen bu iki araştırma konusunun bazı kesitlerinden esinlenerek geliştirilen ve kullanıcının ağ servislerini bulmasını kolaylaştıran akıllı bir sistem geliştirilmiştir.

Abstract

Intelligent Semantic Web Service Finder (ISWSF) is a mechanism that discovers Semantic Web Services satisfying client requirements. The increase in web services and lack of semantic base in search mechanisms of UDDI make it difficult for clients to find required web services. We developed a system which uses a Semantic Search Agent (SSA) to discover required web services from web and selects them according to the client requirements then presents them as a result page. The system uses Ontology Web Language for services (OWL-S), which allows semantic description of web services and hence, the Semantic Search Agent is able to understand predefined concepts of Semantic Web Services, extract necessary information and decide on the requirement of a service for a client. The system combines complimentary aspects of two research topics (Smart Web Query Engine and Matchmaking Algorithm of OWL-S/UDDI Matchmaker) to facilitate the provision of web services to a client.

1. Giriş

Günümüzde Evrensel Tanımlama, Keşif ve Tümeleştirme (UDDI)'nin arama mekanizması anlamsal taban yerine, sözdizimsel tabana dayandırılmıştır. Bu makalede, internet ortamında aranan ağ servislerini, bir Anlamsal Arama Ajansı (AAA) yönetiminde arayan, ayrıştırılan servisleri kullanıcıya bir çıktı gibi sunan, anlamsal bir arama mekanizması ile çalışan, akıllı bir sistemden bahsedilmektedir. Geliştirilen AAA, Ontolojilerin ve Anlamsal Ağ Servislerinin (AAS) içeriğini anlayabilmektedir. Bu yeteneği sayesinde, AAS'lere bağlanıp, özelliklerini inceleyerek kullanıcının isteğine uygun bir servis olup olmadığına karar verebilmektedir. Aranılan servise akrabalık/benzerlik seviyesi derecelenerek belli bir yüzdelik değeri hesaplanmaktadır. Sistemin amacı; uygun AAS'leri kolaylıkla bulmak ve bunu ağ servislerini anlamsal dizinde inceleyerek yapabilmektir.

Sistem geliştirilirken, geleneksel ağ servisi teknolojilerinden UDDI ve Anlamsal Ağ Teknolojileri'nden (*Semantic Web Technologies*) OWL-S (*Ontology Web Language for Services*), OWL (*Ontology Web Language*) ve uygulama alanı olarak Java platformu kullanılmıştır.

Geliştirilen sistemin bölümleri, AAS ve onların OWL-S profil/özelliğ dosyaları, AAA, bağlam ontolojileri veri tabanı (OWL-DB) ve UDDI'dan oluşmaktadır.

AAS; geleneksel ağ servislerinin içeriklerinin ve işlevlerinin anlaşılmasına olanak sağlamaktadır. Böylece ağ servislerinin, otomatik olarak keşiflerini yapabilmek, ağ servislerini birleştirebilmek, çalıştırabilmek ve izlemek bilgisayar tarafında mümkün olabilecektir.

Ontolojiler; belli bir alanın (*domain*) biçimsel betimlenmesi için geliştirilmiş anlamsal ağ teknolojisidir. Bilgisayarların da anlayabileceği şekilde alanların sınıflarını (*classes*), alt sınıflarını

(subclasses), özelliklerini (properties), ve işlevlerini (functions) tanımlayabilmemizi sağlamaktadır.

OWL-S (Ontology Web Language of Services); geleneksel ağ servislerinin girdi/çıkıtlı bilgilerini, işlevlerini (processes) ve bulunduğu adres ya da kimlik bilgilerini içerirler. Kısacası, ağ servislerine anlamsallık katan en önemli anlamsal ağ teknolojisidir.

AAA, (Anlamsal Arama Ajanı); geliştirdiğimiz sistemin en önemli parçasından biri olmakla beraber, AAS'lerin içeriklerini, özelliklerini kolaylıkla algılayıp, kullanıcının aradığı servisin niteliklerine uygun olup olmadığına karar veren mekanizmadır.

Geliştirilen sistem, kullanıcının girdiği, girdi-çıkıtlı bilgileri ile, incelenen AAS'lerin girdi-çıkıtlı bilgilerini, anlamsal tabanda akrabalıklarının/benzerliklerinin belirlenmesi işlemini, ontoloji veritabanından temin etmektedir.

Günümüzde kullanılan ağ servisi teknolojilerinden, UDDI ve WSDL gibi standartların, anlamsal parçalar içermemesi ve ağ ortamında ağ servislerinin sayıca artışı, ağ servis arayıcıları için doğru ağ servislerini bulabilmesini zorlaştırmıştır. Bu sorundan dolayı, ağ'ın akıllı bir sisteme ihtiyaç duyması anlamsal ağ'ı akıllara getirmiştir. AAA, kullanıcı aradığı uygun ağ servisini, kullanıcının girdiği bilgilere bağlı olarak seçmektedir. Bunun için AAS'lerin anlamsal tanımlamalarından, girdi/çıkıtlı bilgilerini, işlevlerini ve bulunduğu adres ya da kimlik bilgilerini incelemektedir.

Öncelikle ajan, kullanıcıdan anahtar kelimeleri almakta ve bunları ontoloji veritabanında arayarak hangi alan ontolojilerinin bu kelimeleri içerdiği belirlenmektedir. Belirlenen alanlar kullanıcıya bildirilir ve bir sorgu yardımıyla, hangi alan/ontoloji ile ilgilendiği sorulur. Kullanıcının ilgilendiği alan belirlendikten sonra, sadece o alana ait ontoloji(ler) üzerinde çalışılır. Kullanıcının verdiği anahtar kelimenin, seçilen ontoloji üzerinde eşanlamlı (Synonym) ve bir örneği (Is_a) bağlantılı terimler tespit edilir, böylece arama terimleri çoğaltılmaktadır. Daha sonra tüm ilgili bulunan terimler, UDDI' ya tüm ilgili ağ servislerinin adreslerini ve bilgilerini elde edebilmek için gönderilmektedir. Daha sonra ajan, yakalanan tüm ilgili ağ servislerinin, OWL-S dosyalarına bağlanarak, tüm gerekli girdi/çıkıtlı değerlerini ayrıştırılmaktadır. Anlamsal Arama Ajanı, kullanıcının girdi/çıkıtlı değerleri ile her ağ servisinin girdi/çıkıtlı değerlerini (AAS'lerdeki anlamsal tanımlamalarında ayrıştırılan profil bilgileri) karşılaştırarak değerlendirmektedir.

Daha önceki makalemizde [5] önerisini yaptığımız bu sistemi Akıllı Anlamsal Ağ Servis Tarayıcısı (Intelligent Semantic Web Service Finder-ISWSF) diye adlandırdık. Bu sistemin geliştirilmesi, uygulanmasını ve elde ettiğimiz sonuçları bu bildiriye topladık. Bu sistem, iki ana araştırma konusunu bünyesinde barındırmaktadır: Akıllı Ağ Sorgu Motoru [6] ve Uyum Algoritmasıdır[7-9].

Bu makale aşağıdaki sıra ile sunulacaktır: bölüm 2 de geliştirilen sistem ve uygulanan metotlar, bölüm 3 de sistemin çalışma sırasını gösteren bir işlev grafiği,

ayrıca sistemin çalışma mekanizması hakkında iki önemli işlevden bahsedilecektir ki bunlar sırayla: Girdi Terimlerinin Anlamsal Çoğaltılışı ve Uyum Görevidir. Bölüm 4, bir örnek çalışma ve sonuçlarından bahsetmektedir. Bölüm 5, ilgili araştırmalar ve sonuçları içerecektir.

2. Akıllı Anlamsal Ağ Servis Tarayıcısı ve Kullandığı Metotlar

Geliştirdiğimiz Akıllı Anlamsal Ağ Servis Tarayıcısı (Intelligent Semantic Web Service Finder-ISWSF) sistemi iki önemli parçadan oluşmaktadır. Birincisi, Girdi Terimlerinin Anlamsal Çoğaltılışı ve ikincisi ise Uyum Görevi'dir. Akıllı anlamsal ağ servis tarayıcısındaki 'Akıllı' teriminden kasıt, sistem, kullanıcı tarafından girilen anahtar kelimeyi sözdizimsel tabanda algılama yerine, anlamsal tabanında algılanmaktadır. Diğer bir deyişle, girilen terimin anlamı hakkında sistemin bilgisi olmaktadı ve o terimin sözdizimsel benzeri olan fakat anlamsal farklılık gösteren terimleri dışlayıp doğru alan üzerinde arama işleminin yapılması sağlanmaktadır. Örneğin, ses dalgaları ile ilgili bir ağ servisi arandığında, deniz hareketlerinden oluşan dalgalar ile ilgili bir ağ servisi kullanıcıya sunulmamalıdır. Böylece sistem kullanıcının girmiş olduğu anahtar kelimenin hangi alanda olduğunu bulabilmektedir. Bu işlemin amacı arama yapılacak alanı sınırlandırmak ve aynı anahtar kelimeyi bünyesinde bulunduran istenilmeyen alanları veya arama alanlarını ayırtmaktır. Böyle bir sistemin en büyük avantajı, arama esnasında istenilmeyen ağ servislerini ayırtmak ve bundan dolayı oluşabilecek bir zaman kaybını önlemektir. Sistem kullanıcının ilgili olduğu arama alanını bilmektedir. Böylece istenilmeyen ağ servislerini ayırtırken, aranan ağ servislerine ulaşma doğruluk oranı artacaktır.

2.1 İlk görev: Girdi Terimlerinin Anlamsal Çoğaltılışı

Daha önceden belirtimiz gibi geliştirilen sistemin, iki önemli görevinden birincisi, Girdi Terimlerinin Anlamsal Çoğaltılışı ve ikincisi ise Uyum Görevi'dir. Girdi Terimlerinin Anlamsal Çoğaltılışı, ilgili ağ servislerini araştırmak için UDDI' ya göndereceğimiz anahtar kelimelerinin sayısını çoğaltmak için gereklidir. Örneğin, kullanıcının uygun ağ servisi aramak için girdiği anahtar kelime 'Araba' terimi olduğunu düşünelim. Bu anahtar kelimeyi tek başına UDDI' ya gönderdiğimizde elimize sadece 'Araba' kelimesini içeren ağ servisleri gelecektir. Yani 'Araba' kelimesinin eşanlamlısı olan 'Otomobil' veya bir örneğidir anlamına gelen 'Araç' kelimelerini içeren ağ servisleri iskanlanmış olacaktır. UDDI' dan daha fazla ilgili servis ele geçirmek, girilen kullanıcı teriminin akraba/benzer terimlerine bağlıdır. Bu akraba terimlerini, kullanıcının seçtiği ontolojiden (ontoloji veri tabanından) sistem bulabilmektedir. Sistem son olarak, kullanıcının girmiş olduğu anahtar kelime ve toplamış olduğu akraba terimlerin tümünü UDDI deposuna gönderip, ele geçirilen tüm ilgili ağ

servislerinin adres ve kimlik bilgilerini ayrıştırmaktadır.

Girdi Terimlerinin Anlamsal Çoğaltılışı'nın yararı, sistemin sürdüreceği arama alanını büyütmek ve ilgili servisleri ıskaladıktan UDDI' dan ele geçirmektir. Sistem bu detayları Girdi Terimlerinin Anlamsal Çoğaltılışı işlemiyle yürütmekte ve bu işlemi ontolojiler sayesinde yapabilmektedir. Sistem bu noktaya geldiğinde ilk görev tamamlanmış ve Uyum Görevi çalışmaya başlamıştır.

2.2 İkinci görev: Uyum Görevi

AAA, UDDI' dan ele geçirilen tüm servislere sırayla bağlanır ve onların üzerindeki OWL-S dosyalarına girer. Kıyaslama işlemi için gerekli olan servislerin girdi/çıkı bilgilerini ele geçirir. Sistem, kullanıcıdan aradığı servise ait girdi/çıkı bilgilerini alır. Daha sonra tüm servislerin girdi/çıkı bilgilerini ve kullanıcının girdi/çıkı bilgilerini, Uyum Görevi algoritmasını kullanarak kıyaslar. Her servis için bir benzerlik değeri hesaplanmaktadır. Bu görevde en önemli nokta, yine yakın terimleri ıskaladıkmadır. Burada yine ontolojiler devreye girer, kullanıcının verdiği girdi/çıkı terimleri ile her servise ait girdi/çıkı terimleri kıyaslanırken akrabalık derecelerine göre bir benzerlik değeri hesaplanır. Örneğin, kullanıcının uygun ağ servisi aramak için verdiği girdi terimi 'Araba' ve çıkı terimi 'Fiyat' olduğunu düşünelim. Yani kullanıcı araba satışı yapan bir servis aramakta olsun. Bu durumda OWL-S dosyalarında 'Araba' kelimesini girdi bilgisi olarak içeren ağ servisleri gelecektir. Yani OWL-S dosyalarında 'Araba' kelimesinin altkümesi veya üst kümesi olan terimler yine ıskalacaktır. Örneğin, 'Mercedes', 'Otomobil' veya 'Araç' terimleri, 'Araba' terimi ile aynı domain de yer almaktadır ve bunları içeren ağ servisleri, müşterinin aradığı niteliklerinde bir ağ servisi olabilmektedir.

Uyum Görevi, Massimo Paolucci ve grubu tarafından geliştirilen 'Uyum Algoritması' ndan türetilerek uygulanmıştır [9]. Massimo Paolucci ve grubu bu çalışmayı UDDI'ya bir eklenti şeklinde tasarladılar. Fakat UDDI deposu bir standart olduğu için onun yapısını değiştirmek yerine, ağ ortamında çalışan bir sistem üzerine yerleştirmek daha doğru olacaktır. Kullanılan uyum görevi geliştirildikçe veya değiştirildiğinde sisteme entegrasyonu daha kolay olacaktır. Sistem, kullanıcının girdiği girdi/çıkı terimleri ile her servise ait girdi/çıkı terimleri arasında bazı ilişkiler belirleyebilmektedir demistik. Örneğin; AYNI, KAPSANIR, KAPSAR ve BASARISIZ. Sırayla ilişkiler ve değerleri hesaplarken kullanılan katsayılar şöyledir:

- **AYNI- (Exact: 3 puan)**
- **KAPSANMA- (Plug in: 2 puan)**
- **KAPSAR- (Subsume: 1 puan)**
- **BASARISIZ- (Fail:0 puan)**

Bu ilişkilere göre sistem, kullanıcı ve UDDI' dan ele geçirilen tüm servislerin girdi/çıkı terimleri için benzerlik yüzdesi hesaplamaktadır. Uygun ağ

servislerini kullanıcının girdiği bu terimlere bağlı kalarak sunmaktadır.

Benzerlik_Derecesi(K,S) {

Eğer

$K \equiv S \rightarrow bd = AYNI \wedge deger = 3)$

Değilse

$K \supset S \rightarrow bd = KAPSANIR \wedge deger = 2$

Değilse

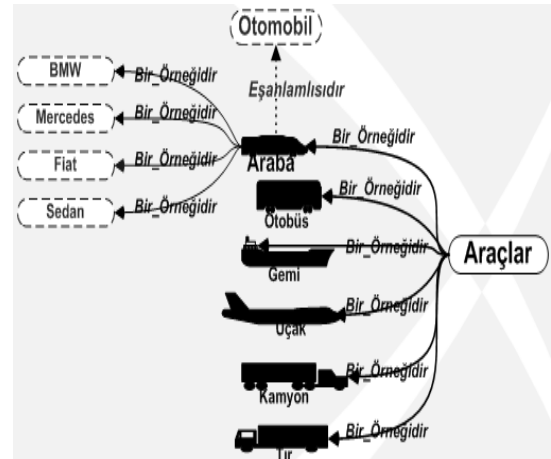
$K \subset S \rightarrow bd = KAPSAR \wedge deger = 1$

Değilse

$K \cap S \rightarrow bd = BASARISIZ \wedge deger = 0 \}$

Yukarıdaki algoritma Uyum Görevi'nin çalışma prensibini anlatmaktadır. Eğer kıyaslanan terimler birbirinin aynısı ise, sistem ilişkiyi AYNI olarak nitelendirecektir ve bu ilişkiye en büyük değeri atayacaktır yani 3 olarak hesaplayacaktır. Eğer kıyaslanan terimlerden, kullanıcının girdiği terim, ağ servisinin girdi teriminin altkümelerinden biri ise, sistem ilişkiyi KAPSANIR olarak nitelendirecektir ve bu ilişkiye ikinci büyük değeri atayacaktır yani 2 olarak hesaplayacaktır. Diğer ilişkiler sırayla yukarıdaki algoritma üzerinde gösterilmektedir.

Örneğin, kullanıcı girdi terimi olarak 'Araba' terimini yazmış olsun. Ağ servisinin OWL-S dosyasında, profil bilgilerinde, saklı olan girdi terimi ise 'Araç' olmuş olsun. Bu durumda 'Araba' terimi 'Araç' teriminin bir alt kümesidir, yani 'Araba' terimi 'Araç' terimi tarafından kapsanmaktadır. Sistem, benzerlik derecesi için KAPSANIR ilişkisini tanımlayacak ve ardından benzerlik yüzdesi hesaplarken bu terimler için değeri 2 olacaktır. Eğer, kullanıcı veya kıyaslanan servisin daha fazla girdi/çıkı terimi var ise, bu durumda, toplam değer puanları ve toplam terim sayısı hesaba katılarak bir yüzdelik derecesi hesaplanacaktır. Eğer, benzerlik yüzdesi %85-%100 arasında ise, bu servis kullanıcının istediği ağ servisinin kendisi ya da çok benzeri nitelikte bir servis olma ihtimali yüksektir.



Şekil 1. Araçlar ilişki diyagramı

Yukarıdaki diyagram Araçlar Ontolojisi'nin bir kısmını şematik olarak göstermektedir.

```

<owl:Class rdf:ID="Vehicle"/>
<owl:DatatypeProperty rdf:ID="Is_a">
  <rdfs:domain rdf:resource="#Vehicle"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Synonym">
  <rdfs:domain rdf:resource="#Vehicle"/>
</owl:DatatypeProperty>
<owl:Class rdf:ID="Car">
  <rdfs:subClassOf rdf:resource="#Vehicle"/>
</owl:Class>
<Vehicle rdf:ID="Car">
  <Synonym
rdf:datatype="&xs:string">Automobile</Synonym>
  <Is_a rdf:datatype="&xs:string">Vehicle</Is_a>
</Vehicle rdf:ID="BMW">
  <Is_a rdf:datatype="&xs:string">Car</Is_a>
</Vehicle>

```

Şekil 2. Araçlar Ontolojisi

Yukarıda ki şekilde, Araçlar Ontolojisi'nin bir bölümü gösterilmektedir. Sistem, bu ontoloji dosyalarına girip, terimler arasındaki akrabalıkları/benzerlikleri kolayca tanımlayabilir. Sistem geliştirilirken, İngilizce dili baz alındığından, geliştirilen tüm ontolojiler de İngilizce olarak yazılmıştır. Yukarıdaki Araçlar Ontolojisi'nin ana sınıfı (Class) 'Araçlar' (Vehicle) terimidir. Bu sınıfa ait iki tür data çeşidi (DataType Property) tanımlanmıştır: eşanlamlı (Synonym) ve bir örneği (Is_a) dir. 'Araba' (Car) terimi, 'Araçlar' (Vehicle) teriminin bir alt sınıfı (Subclass) olduğu ontolojimizde 8.nci satırda gösterilmektedir. 'Car' kelimesinin eşanlamlı terimi olan 'Otomobil' (Automobile) terimi 12.nci satırda gösterilmiştir. Sistem, tüm ontoloji dosyalarına ontoloji veritabanından ulaşabilmektedir.

```

<profile: Profile rdf:ID="Online Car and Parts Selling Service">
  <profile: service Name>Online Car and Parts"></profile: service Name>
  <input><profile: ParameterDescription rdf:ID="Car">
  <profile: ParameterName>Car</profile: ParameterName></input>
  <input><profile: ParameterDescription rdf:ID="Parts">
  <profile: ParameterName>Parts</profile: ParameterName></input>
  <output><profile: ParameterDescription rdf:ID="Price">
  <profile: ParameterName>Price</profile: ParameterName></output>
</profile: Profile >

```

Şekil 3. 'Online Araba ve Parça Satış' servisine ait OWL-S dosyası

Şekil 3, 'Online Araba ve Parça Satış' servisine ait OWL-S dosyasını göstermektedir. Daha önceden de

bahsettiğimiz gibi, OWL-S dosyaları ağ servislerinin üzerinde gömülüdürler. OWL-S dosyaları, servislerin profil bilgilerini, yani girdi/çıkı bilgilerini, işlevlerini ve bulunduğu adres ya da kimlik bilgilerini içerirler. Kısacası, ağ servislerine, bilgisayarlarında ağ servislerinin içeriklerini, özelliklerini kolayca algılayabilmesi için anlam katarlar.

Örneğin, kullanıcının çıktı terimi 'Fiyat' olsun. Şekil 3' deki ontoloji kesitine göre, ağ servisinin OWL-S profil bilgilerinde saklı olan çıktı terimi de 'Fiyat' dır. Bu durumda terimler birbirinin aynı olduğundan, sistem benzerlik derecesini AYNI olarak belirleyecek ve değeri 3 olarak alacaktır.

Sonuç olarak, bizim uyguladığımız anlamsal tabanda arama, bilgisayarın arama alanından haberdar olmasını sağlamaktadır. Bilgisayara 'Ben ne arıyorum?' sorusunu sordurmakta ve olası cevap olarak da 'Ben araba arıyorum!' diye düşündürmektedir. Sistem normal bir arama sisteminden farklılık göstermektedir. Birçok arama sistemlerinde söz dizimsel tabanda arama işlemi yapılmaktadır. Bu nedenden dolayı yanlış ağ servisleri yakalanabilmektedir. Diğer dezavantajı ise, bilgisayar için gereksiz yakalanan ağ sayfalarının neden olduğu gereksiz arama işlemi ve kullanıcı için gereksiz zaman kaybına neden olmaktadır.

3. Akıllı Anlamsal Ağ Servis Tarayıcısı Yapısal Tasarımı

Geliştirilen sistem, ağ kullanıcılarının aradığı en doğru ağ servislerini en kısa zamanda bulmayı hedeflemektedir. Kullanıcının isteğini algılayarak, arama alanını belirler ve zaman kaybına neden olacak arama alanlarını, eleme yoluna gider. Bunları ontolojiler sayesinde yapmaktadır.

Sistem, NETBEANS 4.0 yazılımını kullanarak JAVA platformunda tamamlandı. Ayrıca, sistem üzerinde test edilmek üzere, bazı ontolojiler hazırlandı (örn: Finansal Yatırımlar, Araçlar, Yazılım ve Yapılandırıcılar ontolojileri...). Tüm ontolojiler Protégé 3.1 kullanılarak yaratılmıştır. Yine test uygulamaları için, Protégé 3.1 kullanılarak bazı anlamsal ağ servisleri yaratılmıştır. Sistemin en önemli parçasından biri Anlamsal Arama Ajanı (AAA) olduğu daha önceden belirtmiştik. Ajan arama işleminin başından sonuna kadar kullanıcı ve ağ servisleri arasında bağlantı halindedir ve sistemin beyin bölümünü oluşturmaktadır. İlerleyen bölümler de, ajanın arama işlevine katkıları, ayrıntılandırılmaktadır.

3.1 ISWSF İşlem diyagramı

Şekil 4 geliştirilen sistemin işlem diyagramını göstermektedir. Sistemin çalışma prensibi:

Sistemin İlk Görevi:

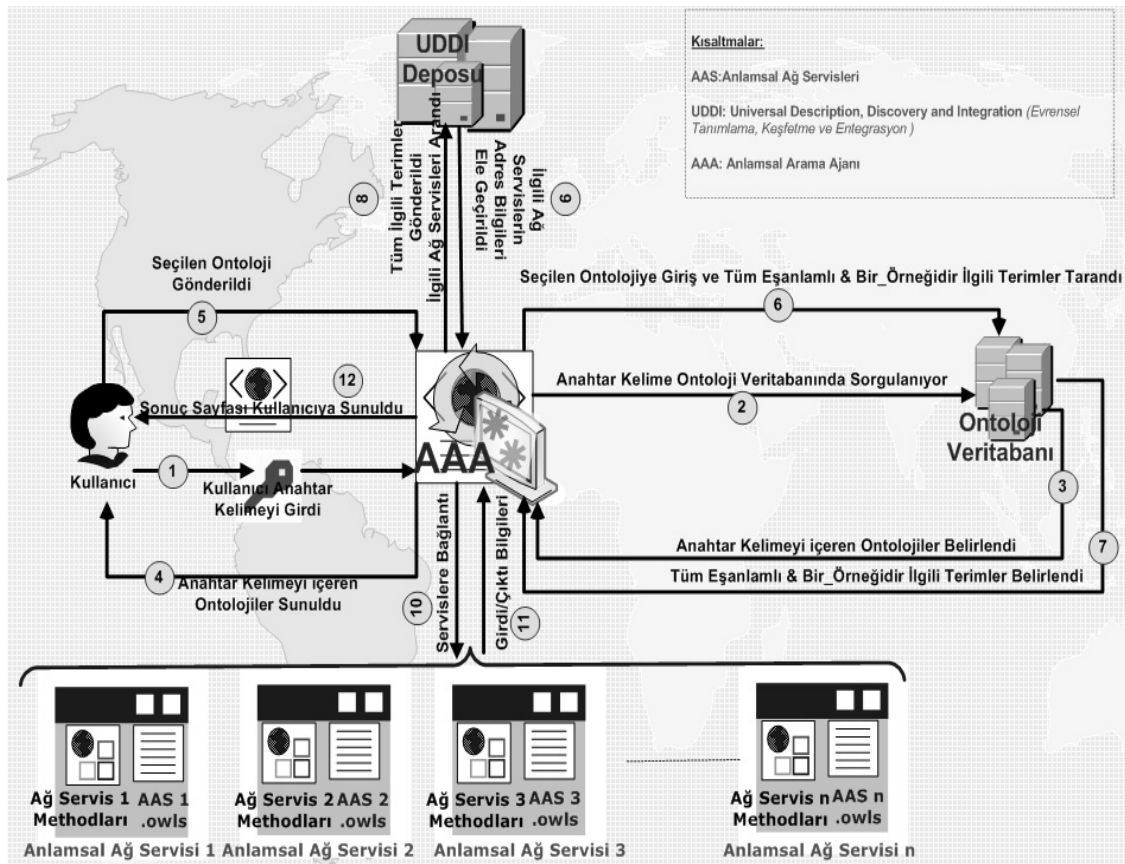
- Başlangıçta, sistem arama alanını belirlemek için, kullanıcıdan anahtar terimlerini girmesini ister (bkz. şekil 4 no 1).

- AAA, ontoloji veritabanında ki tüm ontolojileri aramaya başlar (bkz. şekil 4 no 2).
- Anahtar terimini içeren tüm ontolojiler kullanıcıya bir ara yüz olarak sunulur(bkz. şekil 4 no 4).
- Arama işlemi için en yakın ontolojiyi seçmesi istenir (bkz. şekil 4 no 5).
- AAA artık seçilen ontoloji üzerinde çalışmaya devam edecektir bunun için o ontolojiye bağlanır.
- Seçilen ontoloji içinden anahtar kelimenin eşanlamlı ve bir örneğidir terimlerinin tümü ele geçirilir (bkz. şekil 4 no 6&7).
- Tüm ilgili veya benzer terimler elde edildiğinde UDDI ile bağlantı kurulur ve ilgili tüm ağ servislerinin adres kimlik bilgileri ele geçirilir (bkz. şekil 4 no 8&9).

- Bu noktada sistem ilk görevini, diğer bir deyişle Girdi Terimlerinin Anlamsal Çoğaltılışı işlevini tamamlanmıştır. Sistem ikinci görevi olan Uyum Görevi'ne başlamaya hazırdır.

Sistemin İkinci Görevi:

- Sistem, UDDI' dan elde edilen tüm anlamsal ağ servislere bağlanmaktadır. Tüm servislerin üzerinde gömülü olan OWL-S dosyalarından servislerin profil bilgilerini yani girdi/çıkıtlı bilgilerini okumaktadır(bkz. şekil 4 no 10&11).
- Bu arada sistem kullanıcıya aradığı servis hakkında girdi/çıkıtlı bilgilerini girmesini istemektedir.



Şekil 4. Sistem işlem diyagramı

- OWL-S dosyalarından okunan her girdi/çıkıtlı bilgi kullanıcı tarafından girilen girdi/çıkıtlı bilgileri ile kıyaslayarak benzerlik dereceleri ve değerlere göre her servis için ayrı benzerlik yüzdeleri hesaplanmaktadır. Sonuç olarak, sistem elde edilen tüm benzer servislerin benzerlik yüzdeleri ve bağlanmak için adres linklerini bir sonuç sayfasında

toparlayıp kullanıcıya sunmaktadır(bkz. şekil 4 no 12).

Sistem, bulunan servislerin yüzdelik büyüklüklerine göre EN İYİ, ÇOK İLGİLİ, ORTA veya İLGİSİZ gibi yorumlar eşliğinde sonuçları kullanıcıya sunmaktadır.

4. Deneme Çalışması ve Elde Edilen Sonuçlar

Bu bölümde sistem üzerinde uygulanmış bir örnekten ve elde edilen sonuçlardan bahsedeceğiz. Örneğimizde, kullanıcı 'Araba' terimini anahtar kelime olarak girdiğini ve Araba Satış ve Fiyatları içeriğinde bir ağ servisi aradığını hayal edelim.

4.1 Girdi Terimlerinin Anlamsal Çoğaltılışı İşlemi

Girdi Terimlerinin Anlamsal Çoğaltılışı işleminin amacı, kullanıcının girdiği anahtar kelime yardımı ile çalışma/arama alanımızın sınırlarını belirlemek ve çalışma alanımız ile ilgili ontolojiye girerek anahtar terimin eşanlamlı' larını veya bir_örneğidir bağlantılı terimlerini ele geçirmektir. Böylece, doğru servisleri yakalayıp, kullanıcının girdiği terimin benzer terimleri ile kaydedilmiş ağ servislerini, UDDI deposunda arama yaparken ıskalamamış olacağız. Bunun nedeni ise UDDI üzerinde kullanılan arama tekniği sözdizimsel olarak yani terimlerin yapısına göre yapılmaktadır. Fakat geliştirilen sistem ontolojiler sayesinde kelimelerin anlamlarına bakarak arama yapabilmektedir. Senaryoya göre, sistem kullanıcıya terim giriş ağ sayfasını göndermektedir. Örneğin, kullanıcı 'Araba' (Car) anahtar terimini girmiş olsun. Sistem, bu terimi kullanarak, ontoloji veri tabanının içine girer ve hangi ontolojilerin 'Araba' (Car) terimini içerdiğini tespit eder. Daha sonra sistem, bulduğu tüm ontolojilerin isimlerini kullanıcıya bir ağ sayfası üzerinden göndermektedir. Bu noktada kullanıcı, arama alanına en yakın ontolojinin adını seçecektir.

Şekil 5' de görüldüğü gibi, sistem iki ontolojinin 'Araba' (Car) terimini içerdiğini tespit etmiştir. Kullanıcı, Araba Satış ve Fiyatları ile ilgili bir servis aradığından dolayı, Araçlar Ontolojisi'ni seçtiğini farz ediyoruz. Sistem bundan sonra sadece bu ontoloji üzerinde çalışacaktır.

Şekil 5. 'Araba' terimini içeren iki ontoloji tespit edildi

Sistem, seçilen Araçlar Ontolojisi'ne giriş yapar ve kullanıcının girdiği anahtar terimin tüm ilgili kelimelerini (eşanlamlı, bir_örneğidir) yakalar.

Şekil 6. eşanlamlı ve bir_örneğidir terimleri belirlendi

Şekil 6' da görüldüğü üzere, sistem 'Araba' (Car) anahtar terimi için eşanlamlısı olan "Otomobil" (Automobile) kelimesini yakalamıştır. Ayrıca sistem, 'Araba' (Car) terimi için bir_örneğidir (Is_a) bağlantısı olan 'Araç' (Vehicle) kelimesini yakalamıştır. Gerçekten de 'Araba' (Car) terimi 'Araç' (Vehicle) teriminin bir alt kümesidir ('Vehicle is a car'- bkz. Şekil 1). Tüm bulunan bu ilgili terimler UDDI' ya ilgili ağ servislerini yakalamak için bir sorgu üzerinden gönderilmektedir ve sistem tüm ilgili anlamsal ağ servislerinin adres ve içerik bilgilerini almaktadır. Ayrıca sistem, kullanıcıdan girdi/çıkı parametrelerini istemektedir. Sıradaki işlem, tüm ilgili ağ servislerine bağlanmak ve onların OWL-S dosyalarına girip profil bilgilerini ayrıştırmaktır. Tüm bunlardan sonra, sistem kullanıcının girdi/çıkı parametreleri ve UDDI'da yakalanan ilgili ağ servislerinin girdi/çıkı parametreleri arasında Uyum Görevini çalıştırmaya başlamaktadır.

Şekil 7. Kullanıcı girdi/çıkı terimlerini girdi.

4.2 Uyum Görevi

Kullanıcı, girdi terimi olarak 'Araba' (Car) ve 'Parça' (Parts) terimini ve çıktı terimi olarak da 'Fiyat' (Price) terimini girmiş olsun (bkz. şekil 7).

Daha sonra sistem, ağ servislerinin girdi/çıkıtı parametrelerini, kullanıcının girmiş olduğu bu terimler ile kıyaslamaya başlamaktadır. Önceden de bahsettiğimiz gibi, sistem bu parametreler arasında bazı ilişkiler belirlemektedir. Örneğin; AYNI, KAPSANIR, KAPSAR ve BASARISIZ. Bu ilişkilere göre sistem uygun ağ servislerinin benzerlik oranlarını yüzde olarak hesaplar ve kullanıcıya bir sonuç sayfası içinde gönderir. Sonuç sayfasında, benzerlik yüzdeleri hesaplanan tüm ağ servislerinin kimlik bilgileri ve adres bilgileri sunulmaktadır. Böylece sistem, kullanıcının isteğine bağlı kalarak kullanıcıya hizmet vermektedir.

Örneğimize göre, sistem 'Araba' ve 'Parça' terimleri için, aynısını (Aynı), altkümesini (Kapsar) veya üstkümesini (Kapsanır) içeren terimler yakaladığında, benzerlik derecelerini ve değerlerini belirleyip o ağ servisi için bir benzerlik yüzdesi hesaplamaya başlamaktadır. Sadece girdi terimleri değil çıktı terimleri de aynı işlemde geçmektedir.

4.3 Sonuçlar

Şekil 8' de, örneğimizdeki kullanıcının girdiği 'Araba', 'Parça' ve 'Fiyat' terimleri için, AAA tarafından yürütülen, Uyum Görevi tamamlanmış ve elde edilen sonuçlar gösterilmektedir (Sadece girdi terimlerinin sonuçları gösterilmiştir ve sistem İngilizce tabanlı geliştirildiğinden sonuç sayfasında İngilizce çıktı vermektedir). Kullanıcıya döndürülen bu sayfada, UDDI' dan ilgili servis olarak, dört ağ servisi tespit edilmiş ve sistem tarafından bu hizmetlere ağ üzerinden bağlantı kurulmuştur.

RESULTS
IA.RETURNED MATCH SCORES FOR INPUTS.
1.Semantic Web Service
car->vehicle : Founded Input Relation is= PLUGIN Their Score= 2.0 SWS TotalScore is : 2.0
car->parts : Founded Input Relation is= FAIL Their Score= 0.0 SWS TotalScore is : 2.0
parts->vehicle : Founded Input Relation is= FAIL Their Score= 0.0 SWS TotalScore is : 2.0
parts->parts : Founded Input Relation is= EXACT Their Score= 3.0 SWS TotalScore is : 5.0
INPUTS SCORE WEIGHT: 2.5
2.Semantic Web Service
car->parts : Founded Input Relation is= FAIL Their Score= 0.0 SWS TotalScore is : 0.0
car->car : Founded Input Relation is= EXACT Their Score= 3.0 SWS TotalScore is : 3.0
parts->parts : Founded Input Relation is= EXACT Their Score= 3.0 SWS TotalScore is : 6.0
parts->car : Founded Input Relation is= FAIL Their Score= 0.0 SWS TotalScore is : 6.0
INPUTS SCORE WEIGHT: 3.0
3.Semantic Web Service
car->unit : Founded Input Relation is= FAIL Their Score= 0.0 SWS TotalScore is : 0.0
car->material : Founded Input Relation is= FAIL Their Score= 0.0 SWS TotalScore is : 0.0
parts->unit : Founded Input Relation is= FAIL Their Score= 0.0 SWS TotalScore is : 0.0
parts->material : Founded Input Relation is= FAIL Their Score= 0.0 SWS TotalScore is : 0.0
INPUTS SCORE WEIGHT: 0.0
4.Semantic Web Service
car->time : Founded Input Relation is= FAIL Their Score= 0.0 SWS TotalScore is : 0.0
car->price : Founded Input Relation is= FAIL Their Score= 0.0 SWS TotalScore is : 0.0
car->country : Founded Input Relation is= FAIL Their Score= 0.0 SWS TotalScore is : 0.0
car->date : Founded Input Relation is= FAIL Their Score= 0.0 SWS TotalScore is : 0.0
parts->time : Founded Input Relation is= FAIL Their Score= 0.0 SWS TotalScore is : 0.0
parts->price : Founded Input Relation is= FAIL Their Score= 0.0 SWS TotalScore is : 0.0

Şekil 8. Uyum Görevi algoritması girdi terimlerine uygulandı

Şekilde 8'de, sol tarafta bulunan kelimeler kullanıcının terimlerini ve sağ tarafta bulunan

kelimeler ise ağ servisinin terimleri için kıyaslama yapıldığını göstermektedir. İlk yakalanan ağ servisinin, elde edilen sonuçlara göre, bir AYNI ve bir KAPSANIR ilişkisi bulunmuştur. Buna göre sistem, AYNI ilişkisi için değeri 3 puan ve KAPSANIR ilişkisi için de 2 puan vermiştir. İkinci yakalanan ağ servisinin elde edilen sonuçlara göre, iki AYNI ilişkisi bulunmuştur. Yani kullanıcın girdiği kelimeler ile ağ servisinin girdi terimleri birbirleriyle uyumaktadır. Buna göre sistem tarafından her AYNI ilişkisi için 3 er puan verilmiştir.

İlk yakalanan servisin iki girdi terimi vardır; 'Araç' (Vehicle) ve 'Parça' (Parts) terimleridir. Kullanıcının terimleri ise, 'Araba' (Car) ve 'Parça' (Parts) terimleridir. Her kelime birbiri ile kıyaslanmıştır. 'Parça' (Parts) terimi her ikisinde de olduğundan sistem bu iki terim arasındaki ilişkiyi AYNI olarak değerlendirmiştir. Fakat, 'Araba' (Car) terimi, 'Araç' (Vehicle) teriminin, bir alt kümesi olduğundan, sistem bu iki terim arasındaki ilişkiyi KAPSANIR olarak belirledi ve 2 puanlık bir değer vermiştir.

İkinci yakalanan servisteki girdi terimleri; 'Araba' (Car) ve 'Parça' (Parts) terimleridir. Yani hem kullanıcının hem de bu ağ servisinin girdi terimleri birbirinin aynısıdır ve benzerlik ilişkileri AYNI olarak belirlenmiştir. Sistem her iki AYNI terimleri için 3' er değer puanı vermiştir.

Şekil 9' da, bu iki ağ servisinin benzerlik yüzdeleri sunulmuştur. İlk bulunan serviste, sistem bir AYNI ve bir KAPSANIR ilişkisi yakaladığı için, sistem benzerlik yüzdesini %85 (Input Similarity bölümünde gösterilmektedir) olarak hesaplamıştır. İkinci bulunan serviste iki AYNI ilişkisi yakalandığı için, sistem %100 (Input Similarity bölümünde gözükmemektedir) benzerlik yüzdesi hesaplamıştır.

Tüm hizmetlere ulaşılabilinecek web adresleri ve her servis için benzerlik niteliğini yorumlayan ENİYİ, İYİ, ORTA ve BAŞARISIZ gibi yorumlar sunulmaktadır. Son bulunan servis için sistem, BAŞARISIZ olarak yorum yapmıştır, çünkü, o servis ve kullanıcının girdi bilgileri arasında kesinlikle bağlantı kurulamamıştır.

PERCENTAGES AND ACCESS POINTS OF FOUNDED RELATED SWS
1. SWS is:
Inputs Similarity: 83.333336
Outputs Similarity: 100.0
Average Similarity: 91.666667
THIS SWS ACCESS POINT IS : http://adki.microsoft.com/discovery/businesskey=192386:5-191f-496f-810b-64532dea678
COMMENTS FOR SERVICE: The price of vehicle part offered by Ottawa located company. FINE!
2. SWS is:
Inputs Similarity: 100.0
Outputs Similarity: 100.0
Average Similarity: 100.0
THIS SWS ACCESS POINT IS : http://researchportal.information.com/index.asp?sp=UV&PWVWLVUMV8P&ACUBRQD1hNVL&MR1RnUVwB1HMAA
COMMENTS FOR SERVICE: The lowest and best prices on the web for car audio and car accessories. BEST!
3. SWS is:
Inputs Similarity: 0.0
Outputs Similarity: 100.0
Average Similarity: 50.0
THIS SWS ACCESS POINT IS : http://carb.com/
COMMENTS FOR SERVICE: Online Composite Composite Material Selling
Not GOOD!
4. SWS is:
Inputs Similarity: 0.0
Outputs Similarity: 0.0
Average Similarity: 0.0

Şekil 9. Kullanıcıya sunulan sonuç sayfası

5. İlgili Makaleler

Kaynakça bölümünde sunulan, Akıllı Ağ Sorgu Motoru [6] makalesi, anlamsal ağ metodu üzerine yapılandırılmış bir arama motorundan bahsetmektedir. Ağ üzerinde doğru sayfalara ulaşabilmek için, ağ arama motorları, günümüz İnternet kullanıcıları için önemli bir yer tutmakta ve yaygın olarak kullanılmaktadır. Fakat günümüzde yaygın olarak kullanılan bu ağ arama motorları, anlamsal arama yerine sözdizimsel tabanda arama yaptığından, kullanıcıya gereksiz sayfaları ele geçirmekte ve bu da zaman kaybına ve performans düşüklüğüne yol açmaktadır. Akıllı Ağ Sorgu Motoru, yeni bir sorgu metodunu Anlamsal Ağ Teknolojisi üzerine kurmuştur. Kullanıcının girdiği kelimenin anlamını algılar, daha sonra Google, Yahoo veya Alta Vista gibi ağ arama motorlarında arama yaptıktan sonra, yakalanan sayfalar üzerinde, aynı anlamda veya benzeri olan kaç kelime içerdiğini kontrol eder. Bunun için ontolojileri kullanır. Bu motor, kullanıcının girmiş olduğu anahtar kelime için bir soy ağacı oluşturur. Girilen anahtar kelimenin eş anlamlısı, zıt anlamlısı, bir örneğidir veya etkisidir gibi benzeri kelimeler toplanır. Bu toplanan kelimeler, geleneksel arama motorlarına gönderilir. Sonuçta, ele geçirilen tüm ilgili sayfalar, bazı anlamsal arama filtrelerinden geçirilerek süzgeçlenir. En son kalan ağ sayfaları kullanıcıya sunulur. Doğru ağ sayfalarını bulmaya yönelik yapılan bu sorgu metodunun bir bölümü, geliştirdiğimiz sistemde anlamsal ağ servisleri üzerine uygulandı.

Diğer bir çalışma ise, OWL-S/UDDI Uyum Algoritması' dır [7-10], M. Paolucci ve grubu tarafından geliştirilmiştir. M. Paolucci ve grubu, istenilen anlamsal ağ servislerini, UDDI üzerinden ele geçirebilmek için, UDDI' ya uygulanacak bir eklentiyi önermişlerdir. Çünkü UDDI, ağ üzerindeki ağ servislerinin ilanlarının bulunduğu yerdir ve ağ servislerine ulaşmak ve keşfetmek buradan kolaylıkla yapılabilirdi. UDDI, ağ üzerinde bulunan tüm ağ servislerinin isim, adres veya özellikler gibi bilgileri saklayan bir depo gibidir. UDDI, sözdizimsel olarak ağ servislerinin araştırılabilmesini desteklemektedir.

Uyum Algoritması, UDDI üzerine bulunan servis ilanlarının girdi/çıkıtı_terimleri ve kullanıcının girdiği girdi/çıkıtı_ terimleri parametre olarak almaktadır. Şekil 10' da çıkıtı (*Output-Kısaca out olarak gösterilmektedir*) terimleri için uygulanan uyum algoritması gösterilmektedir.

```
outputMatch(outputsRequest, outputsAdvertisement) {
    globalDegreeMatch= Exact
    forall outR in outputsRequest do {
        find outA in outputsAdvertisement such that
            degreeMatch= maxDegreeMatch(outR,outA)
            if (degreeMatch=fail) return fail
            if (degreeMatch<globalDegreeMatch)
                globalDegreeMatch= degreeMatch
    }
    return sort(recordMatch);}
```

Şekil 10. Çıkıtı terimleri için kullanılan uyum algoritması

Algoritma üzerinde kullanılan terimler, kullanıcının girmiş olduğu çıkıtı terimi 'outR' ve UDDI' deki servisin ilanına ait çıkıtı terimi ise 'outA' olarak gösterilmektedir. Kullanıcı tarafından girilen tüm 'outR' çıkıtı terimleri ile servis ilanında ki tüm 'outA' çıkıtı terimleri arasında yapılan bir kıyaslamadan bahsetmektedir.

Aşağıdaki şekil 11'de, uyum dereceleri için kullanılan bazı kurallardan bahsedilmektedir. Sırayla kurallar 'exact', 'plug-in', 'subsume' ve 'fail' gösterilmiştir. Eğer servisin çıkıtı terimi olan (*outA*) ile kullanıcının çıkıtı terimi olan (*outR*) terimleri aynı ise veya alt kümesi ise, ilişki 'exact' olarak düşünülmüştür. Eğer kullanıcının çıkıtı teriminin (*outR*) alt kümelerinden biri, ilanın çıkıtı terimi (*outA*) ise, ilişki 'plug-in' olarak düşünülmüştür. Tam tersi durumunda 'subsume' olarak nitelendirilmiş ve eğer hiç bir benzerlik bulunamamış ise, bu durumda da 'fail' ilişkisi düşünülmüştür. Bizim geliştirdiğimiz sistem bu algoritmadan esinlenerek üretilmiş fakat küçük farklılar gösteren yeni bir algoritma ile çalışmaktadır (bkz. Bölüm 2.2).

```
degreeOfMatch(outR,outA):
    if outA=outR then return exact
    if outR subclassOf outA then return exact
    if outA subsumes outR then return plugIn
    if outR subsumes outA then return subsumes
    otherwise fail
```

Şekil 11. Çıkıtı terimleri için kullanılan uyum kuralları

6. Sonuç

Ağ servislerinin sayıca artışı ve anlamsal bir arama yapılamaması yüzünden kullanıcının isteğine uygun ağ servislerinin UDDI deposundan elde edilmesi oldukça zordur. Bu sebeple sorunu çözebilmek amacıyla, Akıllı Anlamsal Ağ Servis Tarayıcısı (*Intelligent Semantic Web Service Finder-ISWSF*) diye isimlendirdiğimiz bir sistem geliştirdik. Geliştirilen bu sistemin en önemli parçalarından biri, Anlamsal Arama Ajanı'dır (AAA). AAA'nın amacı, UDDI ile bağlantı kurmak ve ilgili ağ servislerini taramak, anlamsal ağ (Semantic Web) ölçütlerine dayanarak, kullanıcının isteğine en uygun ağ servislerini seçmektir. Geliştirilen sistem, AAA'nın seçtiği servislerin, aranan servise benzerlik-yakınlık derecesine göre yüzdelik değeri hesaplamaktadır. Hesaplanan en büyük yüzdelik değer, en benzer ağ servisi olarak nitelendirilmektedir ve kullanıcının hizmetine sunmaktadır.

AAA daha önceden oluşturulmuş anlamsal ağ servis'lerinin anlam (*OWL-S dosyaları*) dosyası içinde saklı olan profil/özellik bilgilerine girerek ve gerekli bilgiyi oradan ayırıştırarak, kullanıcının isteğine uygun bir ağ servis olup olmadığı hakkında karar verebilmektedir. AAA, kullanıcı ve anlamsal ağ servisleriyle sürekli iletişim kurabilmektedir.

Sistemin omurgası, iki farklı araştırma konusunun birleşiminden yararlanılarak oluşturulmuştur: Akıllı Ağ Sorgu Motoru ve Uyum Algoritması. Bu iki

araştırma konusunu kapsayarak kullanıcıya ağ servislerini bulmasını kolaylaştıran akıllı bir sistem meydana getirilmiştir.

Sistem, kullanıcının verdiği girdi/çıkı terimleri ile her servise ait girdi/çıkı terimleri arasında AYNI, KAPSANMA, KAPSAR veya BASARISIZ gibi isimlendirilmiş ilişkiler belirlemektedir. Sistem, kullanıcının girdi/çıkı terimleri ile her servise ait girdi/çıkı terimleri kıyaslayarak benzerlik dereceleri ve değerleri elde etmektedir ve bu değerlere göre her servis için ayrı ayrı benzerlik yüzdeleri hesaplamaktadır. Sonuç olarak sistem, elde edilen tüm benzer servislerin hesaplanan benzerlik yüzdeleri ve bağlanmak için adres linklerini bir sonuç sayfasında toparlayıp kullanıcıya sunar. Sistem, bulunan servislerin yüzdelik büyüklüklerine göre EN İYİ, ÇOK İLGİLİ, ORTA veya İLGİSİZ gibi yorumlar eşliğinde sonuçları kullanıcıya bir ağ sayfası ile sunmaktadır.

Sistemin en büyük avantajı geliştirilmeye açık olmasıdır. Yani daha doğru servisler yakalamak ontolojiler ile mümkündür. Bu sistemde, sadece iki çeşit benzer kelime (*eş anlamlı & bir örneğidir*) üzerinde arama yapıldı. Eğer daha fazla kelime benzerlikleri kullanılsa (*örneğin bir kelime başka bir kelimenin zıt anlamı, sonucu, etkisi veya kısaltması gibi ilişkiler yaratılarak*) doğru servisi yakalama oranı/verimlilik daha yüksek olacaktır.

Ağ servislerinin en önemli parçaları; profil (properties)/özellik bilgileri ve onların işlev/fonksiyon (process) bilgileridir. OWL-S dosyaları, ağ servislerinin hem profil (properties) hem işlev (process) ve hem de yer/adres (grounding) bilgilerini içermektedir. Bu sistemde, ağ servislerinin sadece profil bilgilerine dayanarak bir arama mekanizması geliştirilmiştir. Eğer sistemin, bu bölümüne eklenti olarak, hem profil bilgilerine hem de işlev bilgilerine dayanarak bir arama tasarlanırsa ve kullanıcının aradığı işlevler/fonksiyonlar bilinirse, istenilen özellikte ve istenilen işlevleri içeren ağ servisleri, ağ üzerinde kolaylıkla yakalanabilir.

7. Kaynakça

[1] <http://www.w3.org/TR/soap/>

[2] <http://www.w3.org/TR/wsdl.html>

[3] <http://www.uddi.org/>

[4] <http://owlseditor.semwebcentral.org/>

[5] **D. Çelik and A. Elçi**, 2005. A Semantic search agent approach: Finding appropriate semantic Web services based on user request term(s), Enabling Technologies for the New Knowledge Society, Proc.: IEEE Publ. ISBN: 0-7803-9270-1, ITI 3rd International Conference on Information & Communication Technology (ICICT 2005), Cairo, Egypt. Pages: 675-689.

[6] **Roger H.L. Chiang, Cecil Eng Huang Chua and Veda C. Storey**, 2001. A smart web query method for semantic retrieval of web data, Data & Knowledge Engineering, Volume 38, Issue 1, Pages63-84.

[7] **K. Sycara, M. Paolucci, A. Ankolekar and N. Srinivasan**, 2003. "Automated Discovery, Interaction and Composition of Semantic Web services", in *Journal of Web Semantics*, Volume 1, Issue 1, pp. 27-46.

[8] **N. Srinivasan, M. Paolucci and K. Sycara**, 2004. "Adding OWL-S to UDDI, implementation and throughput", *First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004)* 6-9, San Diego, California, USA.

[9] **Massimo Paolucci, Takahiro Kawamura, Terry R. Payne, Katia Sycara**, 2002. "Importing the Semantic Web in UDDI", in *Proceedings of Web Services, E-business and Semantic Web Workshop*.

[10] **T. Kawamura, J. A. De Blasio, T. Hasegawa, M. Paolucci, and K. Sycara**, 2003. "A Preliminary Report of a Public Experiment of a Semantic Service Matchmaker combined with a UDDI Business Registry", in *1st International Conference on Service Oriented Computing (ICSOC 2003)*, Trento, Italy.