

Anlamsal Web Tabanlı bir Aktivite Öneri Sistemi Yazılımının Geliştirilmesi

Serdar Korhan Konaray¹, Gizem Aras², Hüseyin Akçekoce², Ömer Faruk Alaca¹,
Moharram Challenger¹, Geylani Kardaş¹

¹ Ege Üniversitesi, Uluslararası Bilgisayar Enstitüsü, 35100 Bornova, İzmir-Türkiye.

² Ege Üniversitesi, Bilgisayar Mühendisliği Bölümü, 35100 Bornova, İzmir-Türkiye.
{korhan.konaray, gizem.aras93, huseyinakcekoce, omerfarukalaca}
@gmail.com, {moharram.challenger, geylani.kardas}@ege.edu.tr

Özet. İnsanlar genellikle boş zamanlarını güzel vakit geçirebilecekleri aktivitelere katılarak değerlendirmek isterler. Uygun olan aktiviteye karar verme süreci için kişisel tercihler, hava durumu gibi çevresel koşullar ve bütçe gibi bazı faktörler önemli rol oynamaktadır. Bu faktörlerin yanında kişinin isteklerine uygun aktiviteleri bulması için bunların tamamını özel bir çaba ile tespit etmesi ve ayıklaması gerekmektedir. Ayrıca, web üzerinde yaptığı aramalarda başarılı sonuçlar elde etmesi için sözdizimsel açıdan tam bir eşleşme olması beklenmektedir. Bu koşul, farklı kategorilendirmelere sahip birçok aktivitenin gözden kaçırılmasına sebep olabilir. Aktivite aramak için ortaya konan bu gereklilikler kullanıcı için zorlayıcı bir ortam oluşturmaktadır. Bu zorluklarla başa çıkmak için, aktivitelerin sağlandığı web servislerini anlamsal olarak tanımlama, uygulamalar aracılığı ile otomatik olarak dinamik bir şekilde keşfedebilme ve kullanıcı tercihlerine göre en yakın aktiviteleri bulma yaklaşımı izlenebilir. Bu amaçla, bu bildiride, anlamsal web servisleri ve anlamsal eşleştirici kullanan bir aktivite öneri sistemi tanıtılmaktadır. Bu doğrultuda, bir mühendislik çözümü olarak, istenen aktiviteyi keşfetmek için web servislerinin iletişim kurabileceği tam teşekküllü bir anlamsal web altyapısı ortaya konmuştur. Önerilen yaklaşımın ve geliştirilen sistemin bir değerlendirmesi de bildiride sunulmuş; yeni yaklaşımın avantaj ve dezavantajları tartışılmıştır.

Anahtar Kelimeler: Anlamsal Web, Web Servis, Ontoloji, Öneri Sistemi, Anlamsal Eşleştirme, Sosyal Aktivite.

Development of a Semantic Web based Activity Recommender Software

Abstract. People are usually willing to spend their free time in entertaining activities. In the process of finding appropriate activity, circumstances such as personal preferences and weather conditions, as well as some factors like the budget play an important role. In addition, one should find and filter these activities with an extensive effort in order to determine the proper activity based on personal requirements. In addition, to succeed in the web-based search for

the activities, exact syntactical mapping is needed. This situation may lead to miss those activities which place in several categories. These requirements for the activity finding make the process difficult for the users. To deal with these challenges, the web services providing the activities can be defined semantically, discovered automatically and dynamically, and hence the activities closest to the user's preferences can be found. To this end, in this paper, an activity recommender system is proposed which utilizes semantic web services and semantic matchmaking. For this purpose, as part of an engineering approach, a semantic web infrastructure is proposed with which web services can communicate to discover the desired activity. The evaluation of both the proposed approach and the implemented system with taking into account the advantages and the disadvantages brought the approach is also given in the paper.

Keywords: Semantic Web, Web Service, Ontology, Recommender System, Semantic Mapping, Social Activity

1 Giriş

İnsanlar genelde boş zamanlarını kendilerine en uygun aktivitelere katılarak değerlendirmek isterler. Aktivite seçimi yaparken ilgi alanlarını, hava durumu ve aktivitelerin konum olarak uzaklığı gibi çeşitli çevresel faktörleri ve bütçelerini de göz önünde bulundurlar. İlgi alanı, kullanıcıların karar verme süreçlerinde büyük rol oynamaktadır, çünkü insanlar eğlenecekleri aktiviteleri seçmek isterler. Diğer taraftan çevresel faktörler de seçimlerini büyük ölçüde etkilemektedir. Örneğin tekne turları hava yağmurluyken yapılamamaktadır ya da kullanıcı hava yağmurlu olduğu için açık hava aktivitelerini tercih etmeme eğilimindedir. Bir diğer önemli özellik ise bütçedir. İnsanlar aktivite seçiminde bulunurken kendi bütçelerine en uygun aktiviteyi seçmek istemektedir. Örneğin ilgi alanı müzik dinlemek olan bir kişi, bütçesine en uygun konserlerin listesine ulaşmak isteyecektir ve eğer hava yağmurluysa açık hava konserlerini tercih etmeyecektir. Fakat bu isteğini gerçekleştirebilmesi için farklı ara yüzlerdeki farklı web servislerine istekte bulunulmalı ve çıkan farklı sonuçlar arasında değerlendirme yapılmalıdır. Bunların yanında web üzerinde arama yapılırken sözdizimsel olarak tam eşleşmenin sağlanması gerekmektedir. Örneğin kullanıcı, konser etkinlikleri ile ilgili bir arama yapıyorsa web servisleri ona tam eşleşme yaklaşımından dolayı konserlerin listesini döndürecektir. Bu da farklı isimlendirmeye sahip benzer aktivitelerin göz ardı edilmesine sebebiyet verebilir. Oysaki o tarihlerde sinemada bir müzikal film oynuyor olabilir ve bu da kullanıcının tercih etmek isteyebileceği bir seçenek olabilir. Tüm bunları göz önüne alırsak web üzerinde en uygun aktivitelerin bulunması ve tahmin edilmesi zorlu bir süreç olmaktadır.

Bu çalışmada Anlamsal Web teknolojileri [1][2] kullanılarak kullanıcı tarafından aktivite bulunması için uygulanan karmaşık süreci daha basitleştirmek ve kullanıcıların karar verme süreçlerini kolaylaştırmak için bir aktivite öneri sistemi gerçekleştirilmiştir. Çalışma kapsamında, bir ontoloji geliştirilmiş, bu ontolojiden yararlanılarak web servislerin anlamlandırılması sağlanmıştır. Anlamsal eşleşme mekanizması, oluşturulan ontolojiden yararlanılarak kullanıcının girdiği kriterlere uygun veya kriterlere yakın sonuçlar elde edebileceği web servislerini otomatik ve dinamik olarak bulmaktadır. Örneğin kullanıcı, JAZZ konserine gitmek istiyorsa ve

JAZZ konserine ait bir sonuç mevcut web servisleri ile sağlanamıyorsa sistem, kullanıcının ilgisini çekebileceğine karar vererek kullanıcıya *POP* müzik konseri önerebilir. Buna ek olarak sistem, ontolojideki verilerden yararlanarak, kullanıcı kriterlerini ve hava durumu gibi çevresel faktörleri de göz önünde bulundurarak aktiviteleri bir puanlama sistemi ile sıraya sokmaktadır. Böylece kullanıcı, zaman kaybetmeden ona en uygun olduğu düşünülen aktiviteyi en üst sırada görebilmektedir.

Bildirinin 2. bölümünde literatürdeki diğer çalışmalar verilmiştir ve bu çalışma kapsamında önerilen çözüm ile ilgili çalışmaların bir karşılaştırması yapılmıştır. 3. bölümde sistem mimarisi açıklanmış; 4. bölümde sistem tasarımı ve önerilen çözüm anlatılmıştır. 5. bölümde uygulama detayları açıklanmıştır. 6. bölümde uygulamanın artı ve eksi yönleri tartışılmış; ve son olarak 7. bölümde elde edilen sonuçlardan ve gelecek çalışmalardan bahsedilmiştir.

2 İlgili Çalışmalar

Literatüre bakıldığında anlamsal web servisler ile gerçekleştirilen ve ontolojilerden yararlanılan birçok çalışma yapıldığı görülmektedir. Örneğin *The Golf Advisor Agent* [3] başlıklı çalışmada *OWL* standardı kullanılarak oluşturulan anlamsallaştırma ile hava durumu da baz alınarak etmen tabanlı karar verme mekanizması gerçekleştiren bir sistem tasarlanmıştır. Benzer şekilde *OWL* (Web Ontology Language) [4] modeline dayanan anlamsal web servisler ile farklı girdi ve çıktılar sağlayan servisler üzerinden acil durumlarda karar verme desteği sunan bir çalışma [5] daha gerçekleştirilmiştir. [6]'daki çalışmada yer alan mimari ise konum bazlı emlak arama işleminin önerilen bir emlak ontolojisi üzerinden anlamsal olarak gerçekleştirilmesini amaçlamaktadır. Bu çalışmalar anlamsal web alt yapısı ve *OWL* standardı kullanımı bakımından bizim çalışmamız ile benzerlik göstermekte, ancak aktivite alanını göz önüne almamaktadır.

Benzer bir çalışma [7] *SAWSDL* standardı ile tasarlanan bir mimariden yararlanılarak web servislerin dinamik keşfi ve çağırımını sağlamaktadır. Sisteme kayıtlı web servisler, kullanılan ortak bir ontoloji ile makineler tarafından otomatik keşfedilmeyi sağlayan anlamsal web servislerine dönüştürülmüş; bu sayede geliştiricilerin statik servis çağrımları için ek çalışma yapmalarına gerek kalmamıştır. Oluşturulan yapı, bir CRM sistemi uygulaması ile denenmiştir. Bu bildiride tanıttığımız sistem yapısı [7] 'deki ile benzerlik gösterse de bu çalışmadan uygulanan yöntem ve uygulama alanı olarak farklıdır.

Aktivite öneri sistemlerine örnek bir çalışma olan [8]'de turistik yerlerin önerilmesi gerçekleştirilmektedir. Bu çalışmada işbirlikçi filtreleme ile toplanan benzer ve farklı verilerin karşılaştırılması sonucu öneri listesi oluşturulmaktadır. Öneri sistemleri çalışmalarına bir başka örnek [9]'da yer almaktadır. Bu çalışmada içeriğe dayalı yaklaşım ile kullanıcının isteklerine ve zamanına bağlı olarak en alakalı aktiviteleri öneren bir sistem geliştirilmiştir. Bizim çalışmamızda, öneri listesi başka kullanıcılardan toplanan bilgilere bakarak veya direkt verilen bilgilere bağlı olarak oluşturulmamaktadır. Bunun yerine kullanıcının bilgileri doğrultusunda *OWL-s* [10] kullanılarak oluşturulan anlamlılaştırmadan yararlanılarak en yakın verilerin elde edilmesi ile bir öneri listesi oluşturulmaktadır.

Hava durumu bazlı aktivite önerme kapsamındaki çalışmalar göz önüne

alındığında [11]'deki çalışmada lokasyon, zaman ve hava durumuna bağlı aktivite önerisi sunan bir yapının tasarlandığı görülmektedir. Çalışmada kullanıcıdan belli parametreler alınmakta; bu parametrelere uyan sonuçlara göre uygun olan aktiviteler listelenmektedir. Her ne kadar bu çalışmanın amacı, bizim önerdiğimiz sistemin amacı ile örtüşse de bu çalışmada herhangi bir anlamsal servis ve ontolojiden yararlanılmamıştır ve sistem sadece geleneksel web servisler ile çalışmaktadır.

Bir diğer çalışmada ise finansal planlama için eşleştirmeye dayalı bir karar destek web hizmeti önerilmiştir [12]. Bu çalışma web servis teknolojileri tabanlı bir model sunmaktadır. Mevcut sorun, bankalar, diğer finans kurumları ve sigorta şirketleri tarafından sunulan çok çeşitli finansal ürünlerin olmasıdır. Belirli bir amaçla ve uygun bir para ile portföy oluşturmak için özelleştirilmiş incelemeler gereklidir ve bu süreçte veri toplama ve ayıklama için çok fazla zamana ihtiyaç vardır. Bu çalışma ile web servis teknolojileri kullanılarak karar verme uygulaması gerçekleştirilmiştir. Kullanıcıdan bazı tercihleri alan uygulama kullanıcıya bu tercihlerden hareketle finansal planlama ile ilgili tavsiyeler vermektedir. Bizim çalışmamızda da benzer şekilde kullanıcılardan tercihler alınmakta ve aktivitelerle ilgili tavsiyeler verilmektedir. Ancak çalışmamızda bu süreç anlamsal web servisleri kullanılarak iyileştirilmiştir.

[13]'teki çalışma ontoloji mühendisliğinin ilkelerini bir işe alım süreci ontolojisinin geliştirilmesi durum çalışması üzerinden aktarmaktadır. Sunulan yaklaşım bu bildiride tanıtılan aktivite ontolojisinin geliştirilmesi sırasında uygulanmıştır.

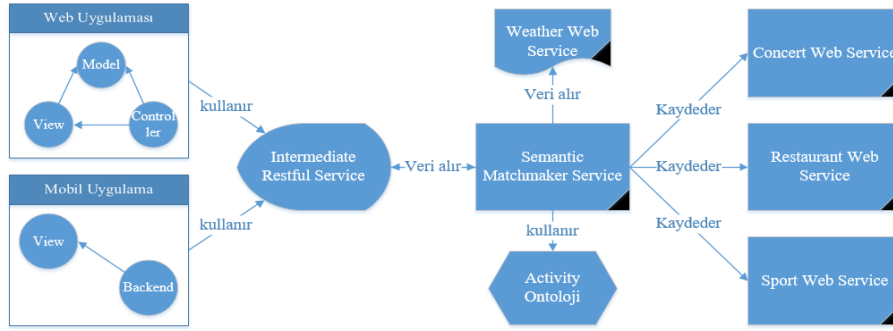
Mevcut çalışmalara genel olarak baktığımızda karar verme mekanizması olarak anlamsal web servis yaklaşımını kullanan birçok çalışmaya rastlanmış, ancak kullanıcı tercihleri ve hava durumuna göre bir karar verme mekanizması sunan aktivite öneri sistemine rastlanmamıştır. Bu bildiride önerilen çalışma ile literatürdeki bu eksikliğin giderilmesi hedeflenmiştir.

3 Önerilen Sistem Mimarisi

Aktivite öneri sistemlerinin bir önceki bölümde bahsedilen sorunlarından hareketle anlamsal bir çerçeveye yerleşebilecek altyapı tasarımı Şekil 1'de görülmektedir. Bu tasarımdaki en önemli bileşen *Semantic Matchmaker Service* bileşenidir. Çünkü bu bileşen çalışmanın önceki bölümlerinde bahsedilen aktivite servislerinin tek tek incelenmesi ve anlamsal olmayan servis altyapılarından dolayı daha az sonuç elde edilmesi sorunlarına çözüm üretmektedir. Ayrıca mimaride dikkat çeken bir diğer nokta ise *Intermediate Restful Service* bileşenidir. Bu bileşen kullanıcılarla etkileşimi sağlayan istemci uygulamaları ile *Semantic Matchmaker Service* bileşeni arasında bir köprü işlevi görmektedir.

Semantic Matchmaker Service bileşeni aktivite arama işleminde tüm süreçte aktif rol oynamakta ve diğer bileşenlerle birebir etkileşim halinde bulunmaktadır. Öncelikli olarak aktivite bilgilerinin sağlandığı servisleri sisteme kaydeden bu bileşen, ayrıca aktivite aramalarında aktif olarak kullanacağı bir *Activity Ontolojisi*ne de sahiptir. Bu bileşenlerin yanı sıra aktivitelerin seçiminde kullanılacak hava durumu gibi diğer bilgiler başka web servislerden alınabilmektedir. Buna örnek olarak bu çalışmada kullanılan *Weather Web Service* verilebilir. Bu servisten alınan hava durumu bilgisi

sayesinde kullanıcının isteklerinin yanında çevresel koşulların da seçim anında etkili olması sağlanmaktadır. Kullanıcıların etkileşim halinde bulunduğu istemci uygulamaları ile iletişimini ise aracı bir restful servis olan *Intermediate Restful Service* sağlamaktadır. Arama isteği geldiğinde sistemine kaydettiği servisleri tarayarak uygun servisi tespit eden *Semantic Matchmaker Service*, dış servislerden ve kullanıcıdan gelen diğer bilgiler doğrultusunda tespit ettiği servisle iletişim kurmakta ve uygun olan aktiviteleri belirlemektedir.

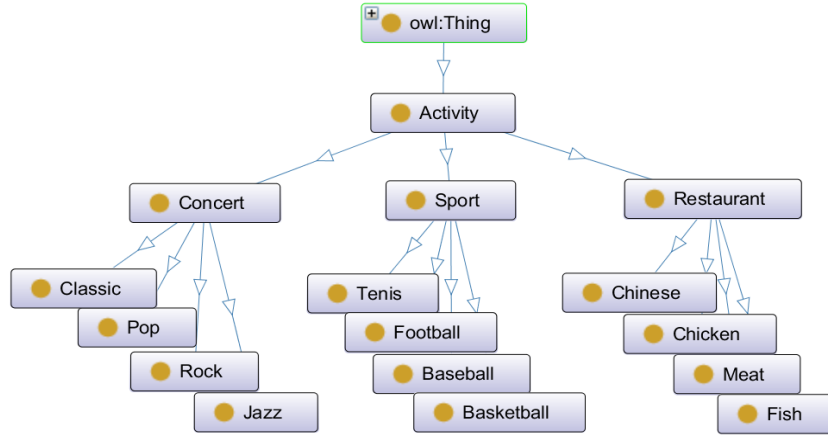


Şekil 1. Aktivite Öneri Sistemi Mimarisi

Mimarinin bir diğer önemli bileşeni ise *Semantic Matchmaker Service* ile istemci uygulamaları arasında köprü vazifesi gören *Intermediate Restful Service*'tir. Bu restful servis istemci uygulamaları ile *Semantic Matchmaker Service* arasında bağımlılığı ortadan kaldırarak tek bir formatta haberleşmelerini sağlamaktadır. Böylece iki taraf için de mümkün olabilecek değişiklikler birbirlerini etkilemeyecektir.

4 Sistem Tasarımı

Bu çalışmada uygun web servislerin öneri sisteminde seçilmesi işlemini anlamsal hale getirmek için web servislerdeki aktiviteleri kategorize eden bir ontoloji tasarlanmıştır. Bu tasarım çalışmasında temel ontoloji mühendisliği ilkelerinden yararlanılmıştır [13]. Tasarlanan ontolojinin bir kısmı Şekil 2'de görülmektedir. Bu ontoloji *Activity* kök düğümünün uzantısı olacak şekilde düzenlenmiştir. *Activity* kök düğümünün altında *Concert*, *Sport*, *Restaurant* gibi kategoriler ve her bir kategorinin altında da kategorileri özelleştiren bir takım alt kategoriler yer almaktadır. Bu ontoloji yardımıyla herhangi bir girdinin ana kategorisi belirlenebilecek, kardeş kategorileri üzerinden benzerlik bazlı sonuçlar elde edilebilecektir.

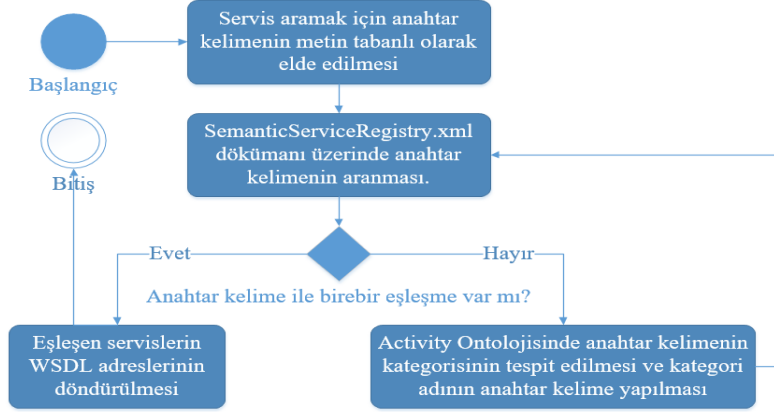


Şekil 2. Aktivite Ontolojisinin bir parçası

Anlamsal çıkarımları yapan ve web servislerinin sistemle bağlantısını yöneten ana bileşen *Semantic Matchmaker Service* web servsidir. Bu servis hem kullanıcılardan aldığı verilerle servis iletişimini kurmakta ve uygun sonuçları kullanıcılara döndürmekte, hem de servislerin anlamsal web servisleri rolü kazanması ve sisteme kaydolmasında doğrudan etkili olmaktadır.

Öneri sisteminin sonuç vermesi adına *Semantic Matchmaker Service*'in ilk görevi sistem için aktiviteleri sağlayan *SOAP* (Simple Object Access Protocol) web servislerinin anlamsal web servisi hizmeti için uygun hale getirilmesi ve kaydedilmesidir. Herhangi bir web servisin anlamsal web servisi işlevi kazanması için *OWL-s* standardına uygun *ServiceProfile*, *ServiceGrounding* ve *ServiceProcess* dökümanlarının hazırlanması gerekmektedir [10]. Bu dosyalar servisin işlevlerini tanımlamakta ve sınırlarını belirlemektedir. Bunun için öncelikle servislerin *WSDL* (Web Services Description Language) adresleri aracılığıyla hangi hizmetleri sunabileceğini belirten *ServiceProfile* dosyaları oluşturulur. Çalışmada ontoloji dili olarak bir W3C standardı olan *OWL* [4] kullanılmıştır. Oluşturulan *ServiceProfile* dosyaları daha sonraki aşamada *Semantic Matchmaker Service* tarafından işlenerek sistemin servisleri kayıt altına aldığı *SemanticServiceRegistry.xml* dökümanına kaydedilir. Bu dökümanda servisin *WSDL* adresi ve ontolojiye uygun olarak sağladığı çıktıların listesi yer alır.

Semantic Matchmaker Service'in ikinci önemli görevi ise girilen anahtar kelimedenden hareketle uygun web servisi tespit etme işlemidir. Şekil 3'te de görüldüğü gibi bu işlem için 2 temel durum ile karşılaşılmaktadır. Birincisi girilen anahtar kelime ile *Semantic Matchmaker Service*'in diğer görevinde bahsedilen *SemanticServiceRegistry.xml* dökümanında yer alan çıktı bilgisinin birebir eşleşmesidir. Bu durumda servislerin *WSDL* adresleri döndürülerek işlem tamamlanır. İkinci durum ise anahtar kelime ile çıktının eşleşmeme durumudur. Bunun neticesinde *Semantic Matchmaker Service* anahtar kelimenin kategorisini tespit etmek için ontoloji üzerinde arama gerçekleştirir, kategoriye tespit eder ve işlemi başa döndürür.



Şekil 3. Semantic Matchmaker Service'e ait aktivite diyagramı

Servislerin tespit edilmesinden sonra ikinci aşama olarak bu servislerden gelen cevapların analiz edilmesi ve puanlanması aşaması gerçekleştirilir. Bu analiz işlemi hava durumu ve aktivitelerin gerçekleştirileceği lokasyonlar baz alınarak yapılır ve *Semantic Matchmaker Service* tarafından uygunluk durumlarına göre aktiviteler puanlanır. Lokasyona göre puanlama işlemi Şekil 4'te örneği gösterilen *City* ontolojisinden yararlanılarak kullanıcı tarafından belirtilen lokasyona komşu olan lokasyonların tespit edilmesiyle gerçekleştirilir. Böylece kullanıcı için yakın olan aktivite sonuçlarının daha ön sıralarda kullanıcıya sunulması sağlanmış olur.



Şekil 4. City.owl Ontolojisi

5 Uygulama

Önceki bölümlerde tasarlanan bu sistem mobil ve web teknolojileri kullanılarak gerçekleştirilmiştir. Mobil istemci için *Android* işletim sistemine uyumlu bir sistem gerçekleştirilirken, web uygulaması için *MVC (Model-View-Controller)* mimarisinde [14] *.NET* teknolojileri kullanılmıştır. Ayrıca birbirinden farklı sağlayıcıların veri sağlama sürecini simüle eden Web servisleri de *Java* programlama dilinde yazılmıştır.

Çalışmanın gerçekleştirim açısından en önemli bileşenlerinden birisi *Semantic Matchmaker Service* bileşenidir ve tasarım bölümünde de anlatıldığı gibi hem web servislerle hem de istemcilerle etkileşim halindedir. Öncelikli olarak web servislerin sisteme anlamsal açıdan kaydedilmesi gerekmektedir. Bunun için web servislerinin *WSDL* dosyalarını kullanarak *OWL-s* standardında anlamsal web servisi dokümanlarının elde edilmesi gerekmektedir. *OWL-s* dokümanlarından bu çalışma kapsamında sadece *ServiceProfile* dosyaları değerlendirilmiştir. *WSDL* dosyasından bu dönüşümün sağlanması için *OWL-s Editor* [15] uygulamasının kütüphaneleri

kullanılmıştır.

OWL-s standardında anlamsal web servisi dokümanlarının elde edilmesi için Şekil 5 ve Şekil 6'daki kod parçaları geliştirilmiştir. Şekil 5'te görülen kod parçası aracılığı ile girdi olarak alınan *WSDL* dokümanı *OWL-s* formatına çevrilip çıkış noktası olarak belirtilen yola kaydedilmektedir. Bu çalışma kapsamında sadece *ServiceProfile* dokümanı kullanıldığı için satır 10'da da görüleceği gibi sadece profile dokümanı ile ilgili işlem yapılmıştır.

```
1 String wsdlPath = args[0];
2 String baseUri = args[1];
3 String type = args[2];
4 String outputFile = args[3];
5 String profileUri = baseUri + "/Profile";
6 WsdlProcessor wsdl = new WsdlProcessor(wsdlPath);
7 OwlsBuilder builder = null;
8 switch (type)
9 {
10     case "Profile": builder = new ProfileBuilder(wsdl);
11     break;
12 }
13 builder.setProfileUri(profileUri);
14 builder.save(outputFile);
```

Şekil 5. *OWL-s* üreten kod parçası

Şekil 5'te gerçekleştirilen işlem için daha önceki bölümlerde açıklandığı gibi *OWL-s Editor* kütüphane dosyaları kullanılmıştır. *WSDL* dosyasından *OWL-s* dosyası elde edebilmek için öncelikle kütüphanenin *jar* dosyası projeye eklenmiş; ardından Şekil 6'daki kod parçası sayesinde argümanları belirlenip çalıştırılmıştır. Böylece herhangi bir *SOAP* servisinin sisteme kaydedilmesi için *WSDL* dokümanını sağlaması yeterli olmaktadır. Gerekli olan dokümanlar *Semantic Matchmaker Service* tarafından otomatik üretilmektedir.

Servislerin sisteme kaydedilmesinden sonraki aşama Şekil 3'te gösterildiği gibi *Semantic Matchmaker Service* bileşeninin kullanıcıdan gelen anahtar kelime doğrultusunda uygun servisi tespit etmesidir. Otomatik servis seçimi için gereken eşleştirmeyi sağlamak amacıyla [16]'daki algoritmadan da yararlanılmıştır. Bu noktada istemcilerin talep ettikleri mekan özellikleri ile aktivite sağlayan servislerde kayıtlı bulunan aktivite özelliklerinin anlamsal eşlenmesi sırasında uygulanan ve [16]'da tanıtılan algoritmayı anlatmakta fayda vardır. İstenen aktivite tipi ve servislerde kayıtlı aktivite tipleri aynı ontolojide bulunmaktadır ve bunlar bir ağaç veri yapısı üzerine yerleştirilebilirler. İstemcinin servis (aktivite) isteğini içeren istek profili *R*, sunucudaki eşleme motoru veritabanına kayıtlanan aktivitelerin profillerinden her biri *A* ile temsil edilecek olursa *A* ve *R* profilleri arasındaki eşleme, öncelikli olarak bir ağaç yapısına yerleştirilmiş aktivitelerin servis tiplerinin karşılaştırılması ile olmaktadır. Eşleştirme başarılarına göre sıralanmış olarak, başlıca dört eşleme düzeyi bulunmaktadır:

- *Tam ("Exact")*: Eğer *A* bildiriminin ontolojik sınıf tipi ile *R* isteğinin ontolojik sınıf tipi birbirine denkse ($A \equiv R$)
- *Uyumlu ("Plug-in")*: Eğer *R* isteğinin ontolojik sınıf tipi *A* bildiriminin ontolojik sınıf tipinin alt kümesi ise ($R \subseteq A$)

- *Kapsayan (Subsume)*: Eğer R isteğinin ontolojik sınıf tipi A bildiriminin ontolojik sınıf tipini kapsıyorsa ($A \subseteq R$)
- *Başarısız (Fail)*: Eğer A bildiriminin ontolojik sınıf tipi ile R isteğinin ontolojik sınıf tipi yukarıdaki durumların dışında kalıyorsa ($A \cap R = \emptyset$)

```

1 public OwlProcessResult StartProcess()
2 {
3     try
4     {
5         var arguments = string.Format("-jar \"{0}\" \"{1}\" \"{2}\" \"{3}\" \"{4}\"",
6             JarFilePath, WsdFilePath, BaseUrl, Type, OutputFilePath);
7         var processInfo = new ProcessStartInfo("java.exe", arguments)
8         {
9             CreateNoWindow = false,
10            UseShellExecute = false
11        };
12        Process proc;
13        if ((proc = Process.Start(processInfo)) == null)
14            return OwlProcessResult.Failed;
15        proc.WaitForExit();
16        int exitCode = proc.ExitCode;
17        proc.Close();
18        return OwlProcessResult.Succesful;
19    }
20    catch (Exception)
21    {
22        return OwlProcessResult.Failed;
23    }
24 }

```

Şekil 6. OwlProcess Kod Parçası

Bu algoritmayı aktivite öneri sisteminde örnekleyecek olursak; *JAZZ* isteğinde bulunan bir istemci için servislerle *EXACT* bir eşleşme gerçekleşmeyecektir. Ancak *PLUG-IN* seviyesinde *JAZZ*'in *CONCERT* aktivitesi ile ilişkisi olduğu tespit edilecektir. Böylece *CONCERT* servisi üzerinden aktivite önerisi getirilebilir. Uygulama aşamasında bu algoritmayı gerçekleştirmek için Şekil 7'de gösterilen kod parçası geliştirilmiştir.

Öncelikle Şekil 7'de yer alan kod parçasında satır 2 ve 3'te *serviceList* ve *keywordList* isimlerinde listeler tanımlanmış ve anahtar kelime listesine kullanıcıdan gelen *keyword* bilgisi eklenmiştir. Ardından *getServicesByKeywordList* metodu ile *keyword* bilgisiyle eşleşen uygun servislerin getirilmesi istenmektedir. Bu metot daha önceki bölümlerde bahsettiğimiz ve Şekil 3'te de gösterilen *SemanticServiceRegistry.xml* dökümanını ayrıştırmaktadır. Bu işlem doğrudan doğruya kullanıcı tarafından girilen kelimenin servislerde karşılığını arama işlemidir. Bu işlemin sonucunda *getServicesByKeywordList* metodunun döndüreceği servisler *serviceList*'i doldurmaktadır. 7. satırda *serviceList* listesinin dolu olup olmadığı incelenmektedir. Eğer bu liste boş ise bunun anlamı herhangi uygun bir servisin bulunmadığıdır. Bu durumda 8. satırda gördüğümüz gibi *semanticSearchByKeyword* metodu çağırılmaktadır. Bu metot *ActivityOntology.owl* ontolojisini kullanarak kullanıcıdan alınan *keyword*'un kategorisini tespit etmekte ve döndürmektedir. Bu işlem sonucunda birden fazla kategori dönebilme ihtimali vardır. Sonuç olarak artık yeni kelimelerin de olduğu bir *keywordList* bulunmaktadır. 9. satırda da görüldüğü gibi tekrar *getServicesByKeywordList* metodu çağırılarak uygun servisler elde edilir. Böylece kullanıcı başlangıçta belirtmemiş olsa bile ontoloji üzerinde yapılan aramalarla kullanıcıya ilişkili sonuçlar döndürülebilmektedir.

```

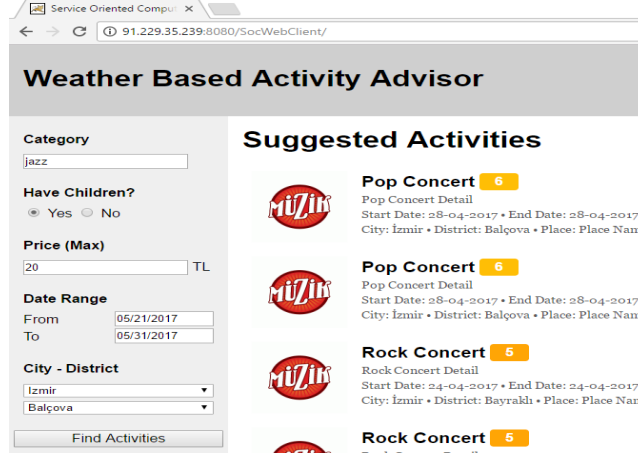
1 public List<String> getServices(String keyword) {
2     List<String> serviceList = new ArrayList<String>();
3     List<String> keywordList = new ArrayList<String>();
4     keywordList.add(keyword);
5     serviceList.addAll(getServicesByKeywordList(keywordList));
6
7     if (serviceList.size() == 0) {
8         keywordList.addAll(semanticSearchByKeyword(keyword));
9         serviceList.addAll(getServicesByKeywordList(keywordList));
10    }
11    return serviceList;
12 }

```

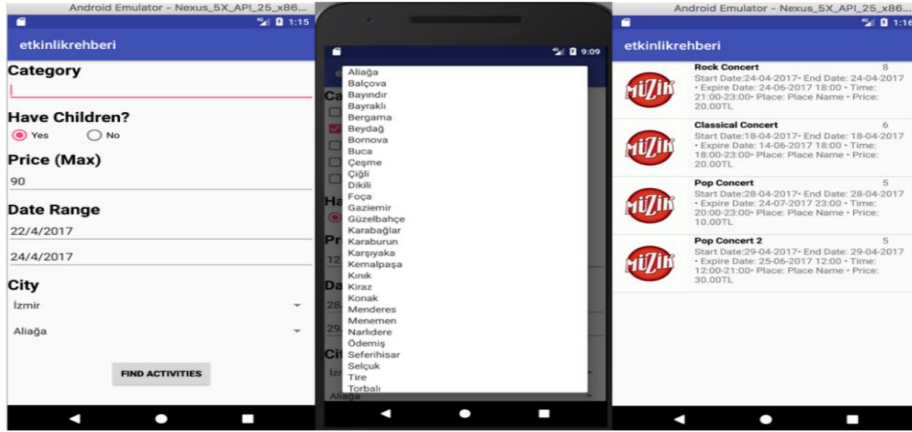
Şekil 7. Semantic Matchmaking Kod Parçası

Çalışmamızda irdelediğimiz bu yakınlık ilişkisiyle ilgili basit bir örneği Şekil 8’de görmekteyiz. Burada kullanıcı *JAZZ* kategorisinde bir aktivite aradığını bildirmektedir. Ayrıca çocuğu olduğunu, 20 TL’den daha fazla bir ücreti aktivite için ayırmak istemediğini, 21 Mayıs – 31 Mayıs 2017 tarihleri arasında bu aktiviteye katılmak istediğini ve son olarak İzmir Balçova lokasyonunda olmasının kendisi için cazip olduğunu belirtmiştir. Bu doğrultuda öncelikle *SemanticServiceRegistry.xml* dökümanında *JAZZ* kelimesiyle uyumlu bir servis aranmış ancak bulunamamıştır. Çünkü Şekil 2’deki ontolojiden de görülebileceği gibi *JAZZ* bir ana kategoriye ifade etmemektedir. Ardından anlamsal yakınlık ile *ActivityOntology.owl* ontolojisi ayrıştırılınca *Concert* kategorisi aramaya değer bir bilgi olarak elde edilmiştir. Bu bilgi *SemanticServiceRegistry.xml*’de arandığında ise *Concert* Web Service aktivite aramak için uygun servis olarak belirlenmekte ve varsa *JAZZ* kategorisinde konser önde olacak şekilde konser aktiviteleri listelenmektedir.

Kullanıcı, sağlanan web arayüzü (Şekil 8) ve mobil arayüz (Şekil 9) ile kendisinden beklenen verileri sisteme girdiğinde *Semantic Matchmaker Service*, kullanıcıdan gelen kategori verisine göre *SemanticServiceRegistry.xml* ve ontoloji verilerini kullanarak servis seçimlerini gerçekleştirir. Servis seçimleri gerçekleştirildikten sonra dönen sonuçlar puanlama algoritmasından geçirilir. Bu algoritmaya göre eğer girilen kategori verisinde tam eşleşme sağlanırsa, eşleşmenin sağlandığı servislere 5 puan eklenir. Tam eşleşme yerine yakın eşleşmenin sağlanması durumunda eşleşmenin sağlandığı servislere 3 puan eklenir. Eğer hava yağmurlu ise kapalı alan aktivitelerine 2, açık ise açık alan aktivitelerine 2 puan eklenir. Kullanıcının seçmiş olduğu ilçe verisinin eşleşmesi durumunda eşleşmenin sağlandığı servise 2 puan daha eklenir. Eğer eşleşme gerçekleşmezse ontolojiden kullanıcının seçmiş olduğu ilçenin komşu ilçelerine bakılır. Eğer komşu ilçe eşleşmesi gerçekleşirse ilgili servislere 1 puan daha eklenir. Tüm aktiviteler, döndürdükleri sonuçlara göre puanlandıktan sonra puana göre azalan sıra ile kullanıcı ara yüzünde listelenir (Şekil 8 ve Şekil 9).



Şekil 8. Web Uygulaması



Şekil 9. Mobil Uygulama

6 Tartışma

Geleneksel web servisler, temel olarak 3 standart üzerine kuruludur. Bunlardan ilki belirli bir web servis için üst veri tanımlayan *WSDL*, ikincisi sunucular ve istemciler arasında veri iletişimini sağlayan *SOAP* üçüncüsü servislerin tanımlanması, keşfi ve entegrasyonunu sağlayan *UDDI* (Universal Description, Discovery and Integration) standartlarıdır. Geleneksel web servisleri, dağıtık sistemler için teknik alt yapı sağlamalarına rağmen anlamsal olarak herhangi bir bilgi içermezler. Bir başka deyişle yukarıda açıklanan standartlardan hiçbiri web servislerin herhangi bir alan açısından ne yaptığını ve web servis metodlarının nasıl bir sonuç ürettiğine dair bir bilgi sunmazlar [17][18]. Dolayısıyla geliştiricilerin, kullanmak istedikleri servisleri bulmaları ve uygulamalarına statik bir şekilde entegre etmeleri gerekmektedir.

Çalışmamızda tasarladığımız *Semantic Matchmaker Service* yapısı sayesinde web servisler otomatik olarak sisteme kayıt olmakta, her web servis de tasarlanan ontoloji yapısına göre anlamlandırılmaktadır. Bu sayede geliştiriciler, web servisleri için vekil sınıfları oluşturmadan ve sisteme yeni servisler kayıt olduğunda herhangi bir ek çalışma yapmadan servislerin kullanımlarını otomatik olarak gerçekleştirmektedir. Ayrıca tasarlanan *Semantic Matchmaker Service* sistemi sayesinde istemciler aradıkları bir aktivite için tam eşleşme bulunmasa dahi aradıkları veriye ait yakın sonuçları da görüntüleyebilmektedir. Bu şekilde bir sistem gerçekleştirimi yapabilmek için geleneksel servis kullanımına ilave olarak *OWL* ve *RDF* gibi ontoloji dilleri kullanılarak sisteme uygun bir ontolojinin tasarlanması ve bu tasarıma göre web servislerin anlamsallaştırılıp bu anlamsallaştırma üzerinden hangi web servislerinin çağırılması gerektiğini yöneten karar verici algoritmaların uygulanması gerekmektedir. Bu da rutin tekniklerden daha karmaşık bir sistem tasarımı anlamına gelmektedir. Ayrıca çalışmada servislerden bağımsız olarak tek bir ontolojinin kullanılması öneri sisteminin yeni servislerle veya etkinlik türleri ile genişlemesi durumunda ontolojinin güncellenmesi gerekliliğini de beraberinde getirmektedir. Karşılaşılan bir diğer zorluk *.NET* ortamında *WSDL* dosyasını *OWL-s* standartlarına dönüştüren bir aracın bulunmamasıdır. Bu çalışmada bu probleme çözüm olarak *JAVA* ile kodlanmış *OWL-s Editor* yazılımı, *jar* dosyası olarak uygulamaya dahil edilmiş ve *ServiceProfile* dosyası oluşturan bölümünü *.NET* içinden çağırıp kullanan bir kütüphane geliştirilmiştir.

7 Sonuçlar ve Gelecek Çalışmalar

Bu çalışmada, geleneksel web servislerin anlaşılması ile kullanıcının tercihlerini ve hava durumunu temel alarak hangi servislerin çağrılacağına çalışma anında karar veren bir aktivite öneri sistemi yazılımı gerçekleştirilmiştir. İstemci tarafı *Android* ve web uygulamaları olarak kodlanmış, karar verme mekanizmasında *OWL-s* standardına uygun oluşturulmuş ontolojiden faydalanılmıştır. Ayrıca, geliştirilen altyapı ile web servislerin anlamsal servislere dönüştürülmesi ve sisteme dahil edilmesi işlemi otomatik olarak yönetilmektedir. Bu sayede uygulama geliştiricileri, web servisleri sisteme dahil olurken herhangi bir işleme gerek duymamaktadır.

Mevcut sistem yapısı, *OWL-s* standardına uygun *ServiceProfile*, *ServiceGrounding* ve *ServiceProcess* dokümanlarından yalnızca *ServiceProfile* dokümanını kullanmaktadır. Sistemde tüm web servisler, atomik olarak çağrılmaktadır. İleriye yönelik olarak mevcut sistemin *OWL-s* standardının *Process Model* yapısı kullanılarak, kompozit servis çağırımları içeren bir çalışma ile desteklenmesi hedeflenmektedir. Ayrıca mevcut sistemde anlamsallaştırma ve eşleştirme, web servislerin çıktı parametrelerine göre şekillenmektedir. İleriye yönelik çalışmalarda *OWL-s* dosyasının girdi ve önkoşul değerleri de dikkate alınıp analiz edilebilir.

Kaynakça

1. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. In: Scientific American, vol 284 (5), pp. 34–43. (2001).
2. Shadbolt, N., Hall, W., Berners-Lee, T.: The semantic web revisited. IEEE Intelligent

- Systems, vol. 21 (3), pp. 96–101. (2006).
3. Athanasiadis, I. N.: Training Intelligent Agents in the Semantic Web Era: The Golf Advisor Agent. 2007 IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol. - Work. WI-IAT Work, pp. 499–502 (2007).
 4. W3C Web Ontology Language (OWL), <https://www.w3.org/OWL/>, Son Erişim 2017/06/21.
 5. Xie, H., Wu, X., Yu X., Li, J., Gu, Y.: Agriculture Emergency Decision System Based on Semantic Web Services, In: International Conference on Convergence Information Technology (ICCIT 2007), pp. 503–507. (2007).
 6. Gümüş, Ö., Kardas, G., Dikenelli, O., Erdur, R. C., Önal, A.: SMOP: A Semantic Web and Service Driven Information Gathering Environment for Mobile Platforms. The 5th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE 2006), October 29 - November 3, 2006, Montpellier, France, Lecture Notes in Computer Science, vol. 4275, pp. 927-940. (2006).
 7. Gümüş, Ö., Yürek, İ.: Anlamsal Web Servislerinin Dinamik Çağırımı. Bilişim Teknolojileri Dergisi, vol. 8, no. 2, pp. 71. (2015).
 8. Jia, Z., Yang, Y., Gao, W. & Chen, X.: User-based Collaborative Filtering for Tourist Attraction Recommendations. In: IEEE International Conference on Computational Intelligence & Communication Technology (CICT), pp. 22-25. (2015).
 9. Mittal, R., Sinha, V.: A personalized time-bound activity Recommendation System. In: IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), pp. 1-7. (2017).
 10. OWL-S: Semantic Markup for Web Services, W3C Member Submission, <http://www.w3.org/Submission/OWL-S/>, Son Erişim 2017/06/21.
 11. Koosam, V.: Weather-based activity advisor. US Patent US7383130 B1 (2008).
 12. Lee, K. K. k., Chiu, D. K., Hung, P. C.: Web-Service Based Information Integration for e-Financial Planning System Matchmaking Decision Support. IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops, pp. 259–262. (2006).
 13. Ahmed, N., Khan, S., Latif, K.: Job Description Ontology. In: International Conference on Frontiers of Information Technology (FIT), pp. 217–222. (2016).
 14. Selfa, D. M., Carrillo, M., Del Rocio Boone, M.: A database and web application based on MVC architecture. In: The 16th IEEE International Conference on Electronics, Communications and Computers, CONIELECOMP 2006, p. 48. (2006).
 15. OWL-s Editor, <http://staff.um.edu.mt/cabe2/supervising/undergraduate/owlseiditFYP/download.html>, Son Erişim 2017/06/21.
 16. Sycara, K., Paolucci, M., Ankolekar, A., Srinivasan, N.: Automated discovery, interaction and composition of semantic web services. Web Semantics, vol. 1 (1), pp. 27–46. (2003).
 17. Ozdil, H., Senkul, P.: Alana Ozgu Web Servis Kesif Sistemleri icin Servis Sorgulama Dili ve Arayuzu, Ulusal Yazilim Muhendisligi Sempozyumu (UYMS 2011), pp 127-133. (2011).
 18. Utku, S., Senkul, P.: Alana Ozgu Web Servis Kesif Sistemlerinde Web Servislerin Test Edilmesi, Ulusal Yazilim Muhendisligi Sempozyumu (UYMS 2012),pp. 107-114, (2012).