

KONUMSAL VERİ ALTYAPILARI İÇİN KONUMSAL PORTAL GELİŞTİRİM ÇERÇEVESİ

M. E. Yıldırım^a, Ç. Cömert^a

^a KTÜ, Karadeniz Teknik Üniversitesi, Harita Müh. Bölümü, Kartografya Anabilim Dalı, Trabzon,
meyildirim@ktu.edu.tr, ccomert@ktu.edu.tr

TUFUAB VII Sempozyumu 2013

ANAHTAR KELİMELER: Konumsal Portal, Servis Yönelimli Mimari, Web Servisleri Kompozisyonu, Web Processing Servisi, Ulusal Konumsal Veri Altyapısı

ÖZET:

Bu çalışmada, dünya genelinde UKVA gerçekleştiriminin en popüler yöntemi olan konumsal portallerin gerçekleştirilmesine yönelik bir çerçeve ortaya koyulmuştur. Bu çerçevenin, hızlı portal geliştirimi, portal ortamında web servisleri kompozisyonunun sağlanması ve açık ve özgür kaynak kodlu portal geliştirimi gibi üç ana bileşeni vardır. Hızlı portal geliştirimi bileşeni, ülkemiz gibi UKVA çalışmalarında geri kalmış ülkelerde konumsal portalin hızla geliştirilmesini sağlayacaktır. Portal geliştiriminin belli standartlar kullanılarak gerçekleştirilmesi ile portal ortamında “yeniden kullanılabilirlik” ve “birlikte işlerlik” sorunlarına çözüm getirilmesiyle hızlı portal geliştirimi sağlanmıştır. Portal ortamında web servisleri kompozisyonu (WSK) bileşeni, kullanıcıların portal ortamında arama, izleme, indirme faaliyetlerinin yanı sıra, daha karmaşık uygulamaları gerçekleştirebilmesine yönelik geliştirilmiştir. Portal teknolojilerinin, servis yönelimli mimaride hala tatmin edici çözüme kavuşmamış olan WSK bakımından sunabileceği katkı incelenmiştir. Portal ortamında, konumsal analizlerin gerçekleştirilmesi için OGC standardı olan Web Processing Servisi (WPS) kullanılmıştır. WPS standardında yer alan WSK yöntemleri sınıflandırması irdelenmiş ve standart için yeni bir sınıflandırma önerilmiştir. Çalışmada ayrıca, mevcut WPS sunucuları incelenmiş ve sunuculardaki çeşitli eksikler dile getirilmiş ve çözüm önerisi ortaya koyulmuştur. Açık ve özgür kaynak kodlu portal geliştirimi bileşeni ile bir konumsal portalin tamamıyla açık ve özgür kaynak kodlu yazılımlar ile gerçekleştirilebileceği ispatlanmış ve bu doğrultuda portal bileşenleri, kullanılacak standartlar ve portal mimarisi belirlenmiştir.

1. GİRİŞ

Konumsal Veri Altyapıları (KVA) birlikte işlerlik altyapılarıdır (Cömert, 2004). KVA'lar, teknolojik ve kurumsal altyapılar olarak iki ana altyapıdan oluşmaktadır. Bu çalışma kapsamında, KVA'ların teknolojik altyapılarına yönelik çalışmalar yapılmıştır. Dünya genelinde KVA çalışmaları incelendiğinde, KVA gerçekleştiriminin en popüler yönteminin, konumsal portaller olduğu görülmektedir. Portallar ve portletler, etkin bir SYM ortamına geçiş sürecinde ara aşama olarak görülmektedir (Akram vd, 2009).

Portal, Java Portlet Belirtiminde, genellikle kişiselleştirmeyi, tek bir oturum açmayı, farklı kaynaklardan elde edilen içerikleri birleştirmeyi sağlayan ve bilgi sistemlerinin sunum katmanlarına ev sahipliği yapan, web tabanlı bir uygulama olarak tanımlanmaktadır (Hepper,2008).

INSPIRE konumsal portal; INSPIRE direktifleri çerçevesinde hazırlanan AB birliği üye ülkelerin konumsal veri setlerine, konumsal veri servislerine ulaşmayı sağlayan, konumsal veri setleri ve servisleri üzerinde ihtiyaca göre arama yapma

imkânını sağlayan web uygulaması olarak tanımlanmaktadır (Inspire, 2005).

OGC (Open Geospatial Consortium) konumsal portal, “veri setleri ve servisler içeren bir çevrimiçi konumsal bilgi kaynakları topluluğu için bir kullanıcı arayüzü” olarak tanımlanmaktadır (OGC, 2004).

Bu çalışma kapsamında, konumsal portallere yönelik bir çerçeve ortaya koyulmuştur. Bu çerçeve, dünya genelindeki konumsal portal çalışmaları sonucu belirlenen eksiklikleri gidermeye yönelik geliştirilmiştir. Üç bileşenden oluşmaktadır. Bunlar; hızlı portal geliştirimi, portal ortamında web servisleri kompozisyonunun (WSK) sağlanması ve açık ve özgür kaynak kodlu portal geliştirimi bileşenleridir.

Konumsal Veri Altyapılarının gerçekleştirilmesinde önemli role sahip olan portallerin hızla geliştirilebilmesi önem arz etmektedir. KVA gerçekleştiriminde geride kalmış ülkeler, bir an önce hızlı bir şekilde konumsal portallerini geliştirmelidirler. Konumsal portallerde geliştirilen web uygulamalarının yeniden kullanılabilir olmaması, her ülkenin ve bu ülkeler içerisinde portal geliştirimi yapan her kurumun, aynı uygulamaları tekrar geliştirmek zorunda bırakmışlardır. Örneğin, INSPIRE

kapsamında geliştirilen portaller için benzer veya aynı uygulamalar tekrar tekrar geliştirilmiştir. Çünkü geliştirilen web uygulamaları yeniden kullanılabilir özellikte değildir.

Ortaya koyulan bu çerçevenin Hızlı Portal Geliştirim bileşeni bu ihtiyacı karşılayacaktır. Bu bakımdan PT'nin sunabileceği olanaklardan biri "portlet yeniden kullanılabilirliği"dir. Portal uygulamalarının mevcut standartlara göre geliştirilmesi, geliştirilen uygulamaların standartları sağlayan herhangi bir portal sunucusunda kullanılabilmesi ve geliştirilmesi portal uygulamalarının hızla gerçekleştirilmesine büyük katkı sağlayacaktır. Bu amaca yönelik olarak bu çalışmada, farklı portal geliştiricilerinin yeniden kullanabilecekleri bazı portletler geliştirilmiştir. Bu portletler, Java Portlet Belirtimi (JSR 286) ve Web Services for Remote Portlets (WSRP 2,0) (OASIS,2008) standartlarına uygun portletlerdir.

Geliştirilen portaller incelendiğinde önemli bir kısmı, merkezi metaveri kataloğuna sahip katalog portalleri olduğu gözlemlenmektedir. Fakat bu yapı günümüzün ihtiyaçlarını karşılayamamaktadır. Dolayısıyla, katalog portallerinden, uygulama tabanlı portallere geçiş yapılması artık zorunlu hale gelmiştir. Uygulama tabanlı portalden kasıt; kullanıcıların konumsal veriyle ilgili uygulamalarını portal ortamında gerçekleştirmesi kastedilmektedir. Konumsal portal kullanıcıları, konumsal analizleri gerçekleştiren web servislerine ve bu web servislerinin zincirlenmesi ile kompleks CBS analizlerin gerçekleştirilebildiği bir portal ortamında ihtiyaç duymaktadır.

Konumsal portal ortamında WSK bileşeni, bu ihtiyacı giderecektir. PT'nin WSK bakımından sunacağı olanaklardan biri olan "Portlet Eventing" mekanizması, portletlerin birbirlerine bağlanabilmesini sağlamaktadır. Portal ortamında WSKyı gerçekleştirmek için kullanılacak web servisleri için konumsal veri üzerinde konumsal analiz imkânını sunan OGC tarafından 2005 yılında tanımlanan ve en son 2007 de güncellenen Web Processing Service(WPS) kullanılmıştır. (OGC, 2007). Ayrıca WSK uygulamaları için WPS'in yanı sıra BPEL standardını (OASIS, 2007) kullanan iş akışı motorları ile BPEL standardını kullanmayan iş akışı motorları da kullanılmıştır. Bunun için, UKVA ortamında uygulama geliştirecek olan kullanıcıların kullanımına yönelik konumsal analiz portletleri geliştirilmiş ve bir örnek uygulama ile bu portletlerin birlikte işleyebileceği gösterilmiştir. Bu çalışmada, web servisleri kompozisyonu için bir örnek uygulama geliştirilmiştir. Örnek uygulama, Çevre ve Şehircilik Bakanlığı'nın ilgili birimindeki bir kullanıcıya yönelik, katı atık deponi alanı yer seçimi analizidir. Bu analizi gerçekleştirmek için WPS işlemleri geliştirilmiştir. Uygulamanın gerek duyduğu verilerin, ilgili kurumlardan Web servisleri ile "anlık" olarak sağlanması öngörülmüştür. Bunun mümkün olamayacağı kurumlar için ise bu yapı uyarlanmıştır.

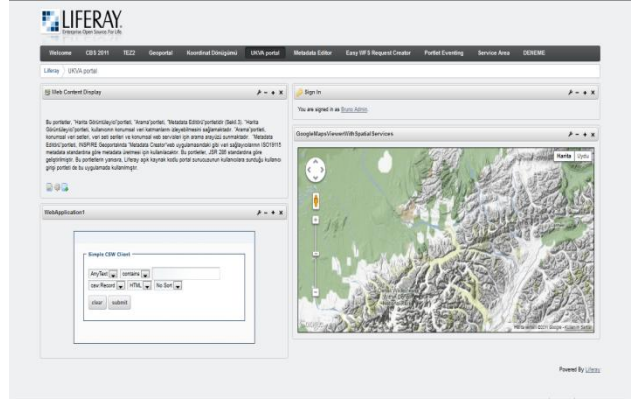
Bu çerçeve kapsamında ortaya koyulan son bileşen Açık ve Özgür Kaynak Kodlu (AÖKK) yazılımlarla portal geliştirim bileşeni; bir portal ve portal bileşenlerinin tamamıyla AÖKK yazılımlarla gerçekleştirilebileceğini ispat etmektedir. Bu doğrultu da portal bileşenleri, kullanılacak standartlar ve portal mimarisi belirlenmiştir.

2. HIZLI PORTAL GELİŞTİRİMİ BİLEŞENİ

2.1. Yeniden Kullanılabilir Portlet Tabanlı Prototip Portal Geliştirimi

Bu çalışmayla, konumsal portallerin geliştirimi için bir yaklaşım ortaya konulmaktadır. Bu yaklaşım kapsamında, "yeniden kullanılabilir" portletler ile prototip konumsal portal geliştirilmiştir. Bu uygulama ile UKVA ortamında, hızlı portal

geliştirimine ihtiyaç olduğu zaman, yeniden kullanılabilir portletler ile nasıl portal geliştirileceği bir portal uygulaması ile açıklanmıştır. Bu çalışmada UKVA Konumsal portaller için üç farklı portlet geliştirilmiştir. Bu portletler, "Harita Görüntüleyici" portleti, "Arama" portleti, "Metadata Editörü" portletidir. "Harita Görüntüleyici" portleti, kullanıcının konumsal veri katmanlarını izleyebilmesini sağlamaktadır. "Arama" portleti, konumsal veri setleri, veri seti serileri ve konumsal web servisleri için arama arayüzü sunmaktadır. "Metadata Editörü" portleti, INSPIRE konumsal portalinde "Metadata Creator" web uygulamasındaki gibi veri sağlayıcılarının ISO19115 metadata standardına göre metadata üretmesi için kullanılacaktır. Bu portletler, JSR 286 standardına göre geliştirilmiştir.



Şekil 1. Prototip Konumsal Portalden ekran görüntüsü

2.2. Portletlerin Yayınlanması

UKVA ortamında birçok kurumda hızlı portal geliştirme ihtiyacı ortaya çıkacaktır. Bir kurumun portal geliştirme aşamasında geliştirdiği uygulamalara, bir başka kurumda portal geliştirimi aşamasında ihtiyaç duyabilecekler. Dolayısıyla, geliştirilen portlet tabanlı web uygulamalarına portal geliştiricilerinin ulaşabilmesi için yayınlanması gerekmektedir. Geliştirilen portletlerin yayınlanması için iki yaklaşım söz konusudur. Bu yaklaşımlardan ilki, portletlerin, portlet kataloglarından paylaşılması, diğeri ise portletlerin WSRP olarak yayınlanması, WSRP portletlerinin WSDL (W3C, 2001) tanımları ile UDDI kataloglarına kaydedilmesi ve ihtiyaç duyan kullanıcının UDDI katalogundan arama yaparak portletleri bulmasıdır. Birinci yaklaşım için açık kaynak kodlu ve ticari portal yazılımlarının kendi portlet katalogları mevcuttur. Portal geliştiriciler bu kataloglardan ihtiyaç duydukları portletlerin dosyalarını indirerek portlet yükleme arayüzünden portal sunucusuna yükleyebilmektedir.

UKVA ortamında kurumlar bazında geliştirilen portlet tabanlı web uygulamalarının paylaşımı için portlet katalogları oluşturulabilir. Bu yapıda ilgili kurumların portal üretme aşamaları şu şekilde olacaktır. İlgili birimdeki kullanıcı portlet kataloglarından arama yapar. İhtiyacı karşılayacak portletleri bulunduğu katalogdan dosyasını indirir.

Portalın portlet yükleme arayüzünden portletleri portale yükler. Portleti portal sayfasında uygun bir yere yerleştirerek portal sayfasını kaydeder ve portal sayfasını oluşturur. Çevre ve Şehircilik Bakanlığındaki ilgili birim, portal geliştirimini gerçekleştirmiş ve portlet katalogunda harita görüntüleyici portletini yayınlamışsa, Orman ve Su İşleri Bakanlığında ilgili birimdeki kullanıcı, bu katalogdan harita görüntüleyici portletini indirebilecek ve portaline ekleyebilecektir. Bu şekilde UKVA ortamında hızlı portal geliştirimi sağlanacaktır.

Bu yaklaşımda, istemci taraf portlet sağlayıcılarının oluşturdukları portlet kataloglarından arama yapacaktır. Bu portlet katalogları herhangi bir standarda göre tasarlanmamıştır. Bu kataloglarda ki portletler, Java Portlet Belirtimine göre hazırlanmış yerel portletlerdir. Kullanıcı ihtiyaç duyduğu portleti bulana kadar, oluşturulmuş portlet kataloglarını inceleyecektir. Bu yaklaşım ihtiyaç duyulan portletin bulunmasını zorlaştırmaktadır.

Portletlerin yayınlanmasında ikinci yaklaşım ise WSRP portletlerinin WSDL dokümanlarının UDDI kataloğuna kaydedilmesi ve kullanıcılarının UDDI kataloğundan arama yaparak portletleri bulmasıdır. WSRP portletlerin sağlayıcısının (producer) metaverisi ile UDDI'ya kaydedilmektedir. Bu metadata da Sağlayıcının adı, tanımı, sağlayıcı hakkında anahtar kelimeler, sağlayıcının WSDL linki gibi bilgiler içermektedir. UDDI kataloğunda arama yaparken portlet sağlayıcısının değil portlet için arama yapılabilmesi için portlet metadatasının yayınlanması gerekmektedir. Kullanıcı bu yaklaşımda, UDDI kataloğundan arama yapacak ihtiyaç duyduğu portleti yayınlanan metadatası sayesinde bulabilecektir (Oracle,2008). Bu yaklaşım portlet aramalarını standartlaştırılmasını sağlamaktadır.

Dolayısıyla, portlet aramaları daha hızlı ve kolay olacaktır. Çalışma kapsamında geliştirilmiş olan yerel portletlerinin ("The Easy WFS Request", "Intersect" ve "Eğim Haritası" portletleri) yayınlanması için ilk yaklaşımdaki gibi portlet kataloğu üretilmiştir. Geliştirilmiş WSRP portletlerinin ("Harita Görüntüleyici" ve "Arama" portletleri) yayınlanması için ikinci yaklaşımdaki gibi WSDL tanımları ile UDDI'ya kaydedilmiştir. Geliştirilen uygulama tabanlı portaldaki portletler arasındaki iletişim için "Portlet Eventing" mekanizması kullanılmıştır (Hepper,2008).

Portal geliştiriminde açık kaynak kodlu yazılımlar kullanılmıştır. Bunlar, OpenLayers¹, Liferay², Geoserver³, Netbeans IDE⁴, Excat CSW sunucusu⁵, Jboss Portal⁶ bu çalışma kapsamında kullanılan açık kaynak kodlu yazılımlardır.

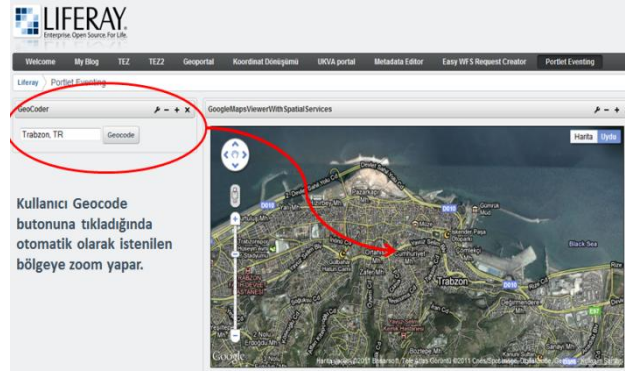
3. PORTAL ORTAMINDA WSK BİLEŞENİ

3.1. "Portlet Eventing" ile Web Servisi Kompozisyonu

WSK, bir kullanıcı tarafından gerçekleştirilmek istenen karmaşık bir uygulamanın, tek bir Web servisi tarafından yerine getirilememesi durumunda, uygun Web servislerinin birleştirilerek uygulamanın gerçekleştirilmesi şeklinde tanımlanabilir (Akıncı, 2006). WSK, SYM'nin asıl katkı noktalarından bir tanesi olduğu için önem arz etmektedir. Bu çalışmanın amaçlarından bir tanesi olan portal ortamında web servisleri kompozisyonunun gerçekleştirilmesi, bu bakımdan önemlidir.

Bu çalışma kapsamında, konumsal portaller için getirilen yeni bir yaklaşım söz konusudur. Portletler içeriklerini, web servislerini kullanarak üretebilmektedir. Dolayısıyla, "Portlet Eventing" mekanizması ile portletlerin birbirleriyle bağlanmasıyla, web servisleri arasındaki iletişimin gerçekleştirilmesi yoluna gidilmiş ve web servislerinin kompozisyonu sağlanmıştır. Buna örnek olarak basit bir uygulama geliştirilmiştir. Bu uygulama, iki web servisi ve iki portlet den oluşmaktadır. Bunlar, "Geocode" ve "Google Maps"

servisleri ve portletleridir. Örnek bir uygulamada kullanıcı, Google Maps portletinde görmek istediği konumun adını girerek (örneğin: Trabzon vb.) "Geocode" butonuna tıklayarak, konuma ait enlem, boylam değerlerini "event" olarak yayımlar ve diğer portlet bu "event" değerini alarak istenilen bölgeye otomatik olarak yakınlaştırma yapar (Şekil 2).



Şekil 2. Portlet Event ile web servisi kompozisyonuna basit düzeyde bir örnek

3.2. OGC Web Processing Servisi ve WSK

Open Geospatial Consortium (OGC), 2005 yılında (OGC, 2005) Web tabanlı konumsal analizlerin, gerçekleştirilmesine yönelik olarak Web Processing Servisi v1.0 (WPS) standardını, 2007 yılında yayınlamıştır (OGC, 2007). Bu standart ile konumsal WPS işlemlerin girdi çıktı verilerini tanımlamayı, işlemlerin nasıl çalıştırılacağını ve çıktı verisinin nasıl tanımlanacağını standart hale getirilmesi amaçlanmaktadır.

WPS servisi, konumsal analizleri gerçekleştirmek için işlemleri (process) kullanmaktadır. WPS işlemlerinden kasıt, konumsal veri üzerinde çalışan herhangi bir algoritma, hesaplama ya da modeldir (OGC, 2007). Bir WPS işlemi, yol verisine buffer uygulayan bir işlem de olabilir, daha karmaşık, yer seçimi analizleri gerçekleştiren bir işlem de olabilir. WPS servisi, vektör ve raster veriler üzerinde işlem yapabilmektedir. WPS servisinin işlem yapacağı konumsal verinin formatında herhangi bir sınırlama yoktur. Kullanılacak konumsal veri, "GeoTIFF", "GML" ya da bir WFS isteğinin yanıtı olan bir yol verisi olabilir.

Bu standart, tüm WPS servisleri tarafından desteklenmesi zorunlu üç operasyon tanımlamıştır. Bunlar; *GetCapabilities*, *DescribeProcess*, *Execute* operasyonlarıdır.

3.2.1. WPS Operasyonları

3.2.1.1. GetCapabilities Operasyonu

GetCapabilities operasyonu; kullanıcının WPS servisinin gerçekleştirebileceği operasyonlar hakkında bilgi veren Capabilities dokümanını sunmaktadır. Capabilities dokümanı, WPS sunucusu tarafından kullanıcılara sunulan işlemlerin isimlerini ve genel tanımlarını da içermektedir (OGC, 2007).

3.2.1.2. DescribeProcess Operasyonu

DescribeProcess operasyonu, WPS sunucusunun sunduğu işlemi kullanabilmesi için gerekli tüm parametreler hakkında bilgi sunmaktadır. Kullanıcı, bu operasyon ile işlemin girdi verileri (inputs), gerçekleştirdiği operasyona dair bilgi, gerçekleştirilecek işlemin desteklediği veri formatları, işlem sonucunda üretilecek çıktı verisi, hakkında ayrıntılı bilgi alır (OGC, 2007). Bu operasyon, hangi işlem hakkında bilgi

¹ OpenLayers, <http://openlayers.org/>

² Liferay, <http://www.liferay.com/>

³ Geoserver, <http://geoserver.org/display/GEOS/Welcome>

⁴ Netbeans IDE, <https://netbeans.org/features/>

⁵ Excat CSW sunucusu, <http://gdsc.nlr.nl/gdsc/en>

⁶ Jboss Portal, <http://www.jboss.org/jbossportal/>

alınacaksa, o işlemin tanımlayıcısı (identifler) kullanılarak gerçekleştirilmektedir.

3.2.1.3. Execute Operasyonu

Execute operasyonu, WPS sunucusu tarafından sunulan işlemin girdi verilerinin tanımlandıktan sonra işlemin çalıştırması ve işlem sonucunda çıktı verilerinin kullanıcıya sunulmasını sağlayan operasyondur (OGC, 2007). Kullanıcı, istediği bir işlemi, tanımlayıcısını (identifler) kullanarak, girdi verilerini ve çıktı verilerini tanımlayarak veri üzerinde işlem yapabilmek için *Execute* operasyonunu kullanır.

3.3. WPS ile WSK

WPS WSK ile kastedilen, herhangi bir WPS de mevcutta tanımlı olmayan, örneğin “katı atık deponi alanı yer seçimi” gibi “yeni” bir işlemin, buffer, intersect gibi temel WPS işlemlerini kullanarak tasarlanması ve/veya gerçekleştirilmesidir. WSK, WPS belirtiminde üç grupta sınıflandırılmıştır ve “Servis Zincirleme (SZ)” olarak anılmıştır. Bunlar, “Business Process Execution Language (BPEL) ile SZ”, “mevcut WPS işlemlerini kendi içinde kullanan yeni bir WPS işlemi tasarlama yoluyla SZ” ve “Key Value Pair (KVP) kodlanmış isteklerle SZ” şeklindedir (OGC, 2007).

Bize göre, WPS standardında WSK yöntemlerinin bu şekilde sınıflandırılması, literatürde kabul görmüş WSK yöntemleri bağlamında kapsayıcı olmamıştır. O nedenle bu çalışmada, Akıncı (2006)'daki WSK sınıflandırmasına bağlı kalmıştır. Buna göre WSK, “tasarım zamanı (design time) tanımlı WSK” ve “koşum zamanı (run time) tanımlı WSK” olarak ikiye ayrılmıştır. Buradaki önemli nokta ise tasarım zamanlı bir kompozisyonun yeniden kullanılabilir olmasıdır. Bu sınıflandırmaya göre, WPS standardındaki KVP SZ, koşum zamanı tanımlı WSK, diğer iki tarz WSK ise tasarım zamanlı WSK grubuna girmektedir.

Diğer yandan, WPS standardında BPEL motoru ile servis kompozisyonu olarak tanımlanan yöntem, kısıtlayıcı olmuştur. BPEL, iş akışı dillerinden (workflow language) sadece biridir. Kullanıcı, BPEL dilini kullanmayan iş akışı motorlarını WPS ile servis kompozisyonu için kullanabilir. O nedenle bize göre “BPEL ile SZ” ifadesi yerine, daha genel bir ifade olan “Bir iş akışı motoru (workflow engine) ile WSK” terimi daha doğrudur. Ayrıca, standarttaki KVP SZ için “basit” servis kompozisyon nitelemesindeki, “basit” terimi, yeterince açık olmamıştır. Standarttaki “basit” WSK'dan kasıt, WSK ile gerçekleşen işlemin basit olması mı, yoksa az sayıda servisin kompozisyonu mudur? Eğer, işlem bakımından basitlik söz konusu ise, ardışık örneğin on adet buffer ve/veya intersect işleminden oluşan bir servis zincirleme, “basit” olarak mı kabul edilmelidir?

3.3.1. Koşum Zamanında Servis Kompozisyonu

3.3.1.1 KVP İsteklerle Az Sayıda Servisin Kompozisyonu

WPS arayüzü kullanılarak servis kompozisyonu yöntemlerinden biri; KVP encoded isteklerle gerçekleştirilen az sayıda servisin kompozisyonu işlemidir. Eğer az sayıda WPS işlemi birbirine bağlanacaksa, bu yöntem kullanılabilir. Çünkü tarayıcıdan sunucuya gönderilecek GET isteğinde karakter sınırlaması 2048 karakterdir. Karakter sayısını aşacak istekler sunucuya GET arayüzünü kullanarak gönderilememektedir. Dolayısıyla, bu karakter sayısını aşacak servis kompozisyonları gerçekleştirilemeyecektir (URL1, 2013). Bu çalışma kapsamında, bu yolla örnek bir veri katmanına ilk önce buffer analizi, daha sonra ise intersect analizi uygulanmıştır. Bu

çalışma için Geoserver WPS ve onun buffer ve intersect işlemleri kullanılmıştır.

Intersect işlemi, buffer uygulanmış yol verisi ve kadastral parsel verisini girdi verisi olarak kullanılmaktadır. WPS sunucusuna gönderilen Intersect isteği:

```
http://localhost:6161/geoserver/ows?service=wps&version=1.0
.&request=Execute&Identifler=JTS:
Intersect&DataInputs=features=@mimeType=text/xml@href=http://
ocalhost:6161/geoserver/tiger/ows?service=WFS&version=1.0.0&requ
est=GetFeature&typeName=tiger:CadastreParcels@Schema=http://lo
calhost:6161/geoserver/schemas/wfs/1.1.0/wfs.xsd,http://localhost:6161
/geoserver/ows?service=wps&version=1.0.0&request=Execute&Identif
ier=JTS:Buffer&DataInputs=features=@mimeType=text/xml@href=ht
tp://localhost:6161/geoserver/tiger/ows?service=WFS&version=1.0.0&
request=GetFeature&typeName=tiger:roads@Schema=http://localhost
:6161/geoserver/schemas/wfs/1.1.0/wfs.xsd&RawDataOutput=buffered
GML@mimeType=text/xml&RawDataOutput=intersectoutput@mimeT
ype=text/xml
```

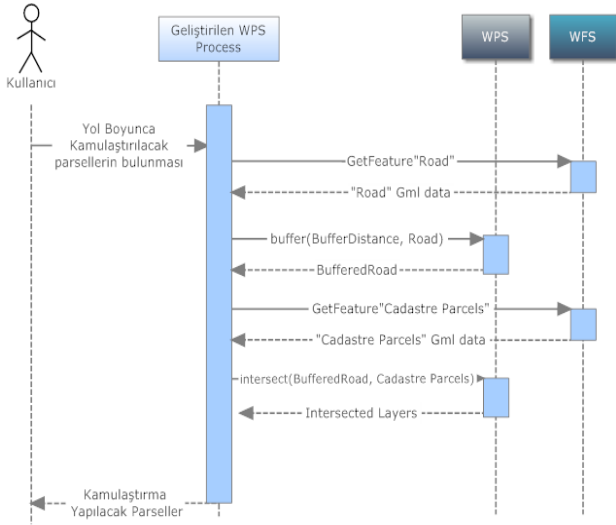
şeklindedir.

Sadece iki basit analiz işleminin kompozisyonu için, yukarıdaki gibi uzun ve karmaşık bir istek oluşturulması gerekiyor. KVP encoded isteklerle, kompleks bir CBS analizinin gerçekleştirilmesi, karakter sınırlamasından dolayı mümkün olmamaktadır. Standartta, KVP encoded isteklerle *Execute* operasyonunun gerçekleştirilmesi seçmeli (optional) dir. Tüm WPS sunucularının, KVP encoded isteklerle *Execute* operasyonunun gerçekleştirilememesi de bu yöntem için bir dezavantaj oluşturmaktadır.

3.3.2. Tasarım Zamanında Servis Kompozisyonu

3.3.2.1. Yeni bir WPS İşlemi Tasarlama ile WSK

Servis kompozisyonunda ikinci yöntem; birden fazla WPS işlemini içerecek şekilde yeni bir WPS işlemi tasarlayarak servis kompozisyonu işleminin yapılmasıdır. Bu yöntemde; Atomik düzeyde işlemleri tasarlamak yerine (örn: buffer, intersect, union vb.), atomik düzey işlemleri kullanarak bütünüyle uygulamayı gerçekleştiren işlemler geliştirilmektedir. Literatürde, bu yaklaşıma yönelik uygulamalar gerçekleştirilmiştir (Stollberg vd, 2007), (Stollberg vd, 2009). Bu çalışma da kullanıcının, yol verisine buffer uyguladığı ve buffer analizi sonucu üretilen veri ile kadastral parsel katmanı intersect yapabildiği, yol projesinde kamulaştırılacak parselleri, kullanıcıya sunan örnek bir uygulama geliştirilmiştir (Şekil 9). Çalışmada, sadece WPS işlemleri birbiriyle kompozite edilmemiş ek olarak, WFS servisi de servis kompozisyonu içerisinde kullanılmıştır. Bu uygulamada geliştirilen WPS işlemi, Deegree WPS ortamında çalıştırılmıştır. Bu tarzda geliştirilen işlemlere “kompozit” işlemler de denilmektedir (Stollberg vd, 2009).



Şekil 3. Geliştirilen WPS "process"inde işlemlerin UML diyagramı

Bu yöntemin avantajları ve dezavantajları mevcuttur. İşlemlerin dağıtık sistemlerde olduğu gibi zincirlenmemesi bu işlemler arasındaki veri paylaşım hızını artırmaktadır. Çünkü analiz tek işlem içerisinde gerçekleştirilmekte ve ağ üzerinden herhangi bir veri transferi söz konusu olmamaktadır. Kapladığı yer bakımından büyük olan verilerle işlem yapılması durumunda, bu verilerin ağ üzerinden servisler arasında aktarılması performansı etkileyecektir. Fakat işlemlerin bu şekilde tasarlanması, geliştirilen işlemlerin yeniden kullanılabilirliğini azaltmaktadır. Bu tarzdaki uygulamalar, UKVA ortamı düşünüldüğünde, kullanılabilirliği sadece hangi uygulamaya yönelik geliştirildiyse, sadece o kesim arasında kullanılabilir. Geliştirilen işlem, sadece analizin gerçekleştirilmesi için kullanılabilir. UKVA gibi dağıtık sistemlerde, farklı geliştiricilerin geliştirdiği servislerin kompozisyonu bu şekilde sağlanamayacaktır.

3.2.2.2 Bir İş Akışı Motoru ile Servis Kompozisyonu

Bir Web servisleri kompozisyonu bir iş akışı (workflow) dilinde ifade edilir. Bu dillerden en popüler olanı WSBPEL dir. WSBPEL-v2.0 standardı, OASIS'in WSBPEL Teknik Komitesi (Web Services BPEL Technical Committee) tarafından, Nisan 2007 yılında güncellenerek son halini almıştır (OASIS, 2007). BPEL standardı, mevcut web servislerini bir arada kullanarak, belirli bir iş amacını gerçekleştirmeyi amaçlamaktadır. XML ve WSDL standartları üzerine kurulmuştur. Bir BPEL işlemi (BPEL process), mevcut Web servislerini kullanarak yeni bir uygulama geliştirmek için, BPEL şemasına uygun olarak geliştirilen bir XML dokümanıdır (OASIS, 2007).

Bu çalışmada, bir BPEL motoru ile WPS WSK gerçekleştirimi yapılmıştır. Bu uygulamada BPEL WSK için, Cape Clear Studio, WPS sunucusu olarak ta Deegree WPS kullanılmıştır. Bir BPEL WSK, kompozisyonun içerdiği Web servislerinin Web Servisi Tanımlama Dili (WSDL- Web Service Description Language) tanımlarını kullandığı için, Deegree WPS' te eksik olan WPS WSDL dokümanı da bu çalışmada üretilmiştir. Ayrıca bu çalışmada Deegree WPS te tanımlı "buffer" ve "intersect" işlemlerinden farklı girdi ve çıktı formatları "buffer" ve "intersect" işlemleri tanımlanmıştır. Aşağıdaki WSDL dokümanı, bu çalışma kapsamında üretilen WSDL dokümanının "mesajlar" ve "port tipi" kısmından bir parçadır.

```
<message name="ExecuteProcess_BufferRequest">
```

```
<part name="DataInputs" element="tns:ExecuteProcess_Buffer"/>
</message>
<message name="ExecuteProcess_BufferResponse">
<part name="ProcessOutputs" element="tns:ExecuteProcess_BufferResponse"/>
</message>
<portType name="DeegreeMEY_PortType">
<operation name="ExecuteProcess_Buffer">
<input message="tns:ExecuteProcess_BufferRequest"/>
<output message="tns:ExecuteProcess_BufferResponse"/>
<fault name="ExceptionResponse" message="tns:ExceptionResponse"/>
</operation> ...
</portType>
```

OGC WPS standardına göre, tüm WPS sunucularının WSDL tanımlarını gerçekleştirmeleri zorunludur. WSDL dokümanı, WPS sunucusunun içerdiği tüm işlemler hakkında tanımlamaları içermektedir. Standartta göre, WPS sunucuları WSDL dokümanını kullanıcılara, <http://hostname/WPSname?WSDL> şeklindeki istek sonucunda sunulmalıdır.

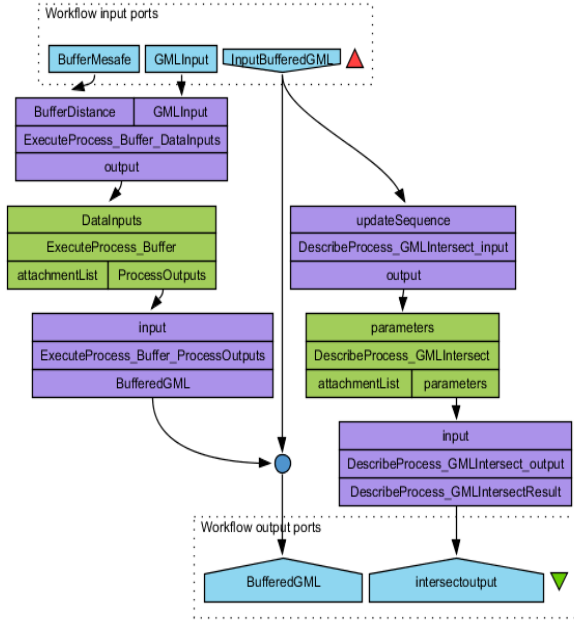
WPS servisi, sadece servisin WSDL dokümanını yayınlamak için bir arayüz sunmaktadır. Kullanıcı, servisin WSDL dokümanını kendi oluşturmak zorundadır. Ayrıca bu standart, WPS sunucularının WSDL dokümanlarını "<http://hostname/WPSname/identifler/service.soap?WSDL>" şeklinde tüm işlemler için ayrı ayrı sunmaları tavsiye edilmiştir. Bu çalışma kapsamında incelenen 52 North WPS, Deegree WPS, Geoserver WPS, pyWPS sunucuları, standartta dile getirilen WSDL dokümanının sunulduğu arayüzü desteklemektedirler. Fakat sunucular WSDL dokümanını otomatik olarak değil, kullanıcıların kendilerinin oluşturmasıyla, sunabilmektedir.

WSDL tanımlarının, kullanıcılar tarafından üretilmesi, oldukça külfetli olmakta ve vakit kaybına sebep olmaktadır. Çünkü bir sunucuda ne kadar işlem varsa, hepsi için ayrı ayrı WSDL tanımları yapmak gerekmektedir. Bu dezavantajı, ortadan kaldırmak için WSDL dokümanları sunucu tarafından otomatik üretilmelidir. WSDL tanımlarının WPS sunucularının Capabilities ve işlemler hakkında ayrıntılı bilgi veren DescribeProcess dokümanından otomatik olarak üretilmesi, bu dezavantajı ortadan kaldıracaktır. Ayrıca, standartta tavsiye edilen işlemlerin tanımlayıcılarına (identifler) göre ayrı ayrı WSDL tanımlarını gerçekleştiren WPS sunucusu mevcut değildir. Dolayısıyla, kullanıcı tüm WPS işlemlerini kapsayan büyük bir WSDL dokümanıya çalışmak zorunda kalmaktadır. Ayrıca, her işlem için ayrı ayrı WSDL dokümanının oluşturulması, servis kataloglarına servisteki tüm işlemlerin tamamının kaydedilmesi yerine işlemler ayrı ayrı kaydedilebilecektir. Böylelikle, kullanıcılara servis kataloglarında işlemlere göre arama yapma imkânı verilecektir. WPS sunucularının WSDL dokümanlarını otomatik olarak üretecek bir araç bulunmadığı için bu çalışmada WSDL editörü olarak, Altova XMLSpy programı kullanılmış ve WSDL dokümanları adı geçen programda elle yazılmıştır.

BPEL motoru ile buffer ve intersect işlemi kompozite edilerek, yeni WPS işlemi tasarlanmasıyla servis kompozisyonu başlığı altında anlatılan uygulama gerçekleştirilmiştir. Uygulamanın bu tarzda geliştirilmesi, geliştirilen işlemlerin yeniden kullanılabilirliğini desteklemektedir. Bu çalışma kapsamında geliştirilen işlemler başka uygulamalarda ayrı ayrı kullanılabilir.

Bu çalışma kapsamında, web servislerinin kompozisyonu için BPEL kullanmayan iş akışı motoru kullanarak servis kompozisyonunu için de örnek bir uygulama gerçekleştirilmiştir. Bu uygulama, Taverna Workbench 2.0 iş akışı motoru kullanarak gerçekleştirilmiştir. Bu yapıda, WPS servisinin WSDL tanımları kullanılarak iş akış şemaları

oluşturulur (Şekil 4). Oluşturulan bu iş akışı, web üzerinden çağrılabilir ve başka iş akış şemalarının içerisinde kullanılabilir.



Şekil 4. Çalışma kapsamında oluşturulan iş akış şeması

4. AÇIK VE ÖZGÜR KAYNAK KODLU PORTAL GELİŞTİRİMİ BİLEŞENİ

Konumsal portalin AÖKK yazılımlarla geliştirilmesinin avantajının ortaya koyabilmek için Açık ve Özgür kaynak kavramlarının iyi tanımlanması gerekmektedir.

Açık Kaynak Girişimi (Open Source Initiative)ne göre açık kaynak; yazılımlar için geliştirme sürecinin şeffaflığının ve farklı gözler tarafından incelenebilmesinin gücünü birleştiren bir geliştirme yöntemidir. Açık kaynak kodu, yazılımların daha kaliteli, daha yüksek güvenilir ve daha düşük maliyetli olmasını sağlayacaktır (Open Source Initiative, 2009).

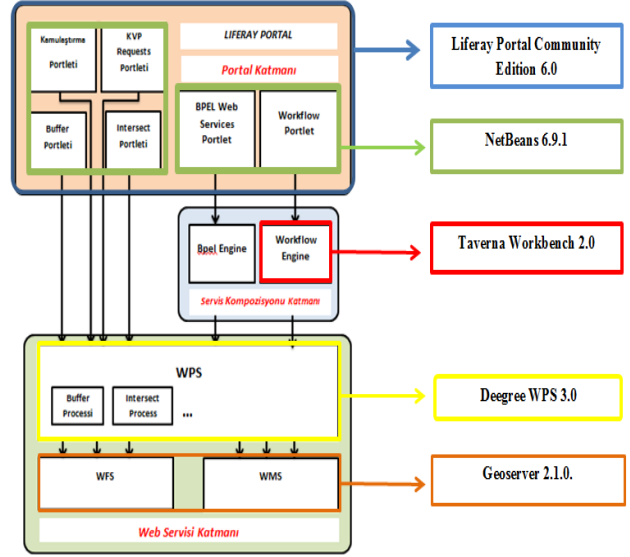
Açık kaynak ile özgür kaynak kavramları birbirine yakın olsalar da tamamen aynı değildir. Literatürde, açık kaynak ve özgür kaynak arasındaki farkların irdelendiği, (URL 2),(URL 3), (URL 4) gibi çalışmalar mevcuttur. Özgür kaynağı tanımlayan Richard Stallman, özgür kaynak olmayan yazılımları sosyal sorun olarak tanımlamakta ve özgür kaynak yaklaşımının da bu sosyal sorunu çözmek için başlatılan sosyal bir hareket olarak tanımlamaktadır. Açık kaynak yaklaşımını ise yazılımın daha güçlü ve güvenilir olmasına yönelik, yazılım geliştirme yöntemi olarak tanımlamaktadır.

Açık ve özgür kaynak kodlu yaklaşımın önemi, dünya genelinde ülkelerin bu yaklaşımı tercih etmesinden anlamak mümkündür. Örneğin; Birleşik Krallık'taki açık kaynak devleti yaklaşımına ekstra destek sağlaması için, devletin tüm seviyelerinde FOSS yazılımlarının kullanılmasını teşvik etmek için, İngiltere hükümeti tarafından taslak bir eylem planı hakkında duyuru yapılmıştır (CIO, 2009).

Günümüzde, UKVA konumsal portallarına yönelik ihtiyacı tam anlamıyla çözebilen ticari yazılım mevcut değildir. Mevcut ürünlerin yeterli olmayışı ve bu yazılımların başka portal ürünleri ile entegrasyonunun sağlanamaması, yeniden kullanılabilirliğinin düşük düzeyde olması gibi sorunlar, konumsal portal geliştiriminin açık ve özgür kaynak kodlu yazılımlarla gerçekleştirilmesinin, doğru bir karar olacağını

göstermektedir. Bu çalışma, bir konumsal portalin tamamıyla açık ve özgür kaynak kodlu yazılımlarla geliştirilebileceğini ispatlaması açısından önem arz etmektedir. Bu çalışmada geliştirilen portal tamamıyla açık kaynak kodlu yazılımlarla geliştirilmiştir. Geliştirilen portal uygulamalarının da, açık ve özgür kaynak kodlu olarak yayınlanması planlanmaktadır.

Portal geliştiriminde açık ve özgür kaynak kodlu yazılımlar; *OpenLayers*, *Liferay*, *Geoserver*, *Netbeans IDE*, *Deegree WPS*, *52 North WPS*, *Geoserver WPS*, *PyWPS*, *Taverna Workbench* bu çalışma kapsamında kullanılan açık kaynak kodlu yazılımlardır.



Şekil 5. Portal Mimarisi ve kullanılan açık kaynak kodlu yazılımlar

Bu çalışmada ortaya koyulan çerçevenin bu bileşeni ile tamamı açık ve özgür kaynak kodlu yazılımlar ile konumsal portal geliştiriminin yapılabilmesinin ispat edilmesi, önemli bir sonuçtur. Bu yaklaşım, konumsal portal geliştiriminin, "hızlı", "kaliteli" ve "az maliyetli" yapılabilmesine olanak sağlaması açısından, önemli bir yaklaşımdır.

5. SONUÇ

Bu çalışmada, konumsal portalların gerçekleştirilmesine yönelik bir çerçeve, ortaya koyulmuştur. UKVA ortamında konumsal portalların nasıl geliştirilmesi gerektiği, ortaya koyulmuş ve bu doğrultuda prototip konumsal portallar geliştirilmiştir.

Hızlı portal geliştirimi yaklaşımı, UKVA ortamında aynı veya benzer uygulamaların geliştirilmesi ve bu uygulamaların başka portal uygulamalarında yeniden kullanılamaması sorununun çözümü için getirilmiş bir yaklaşımdır. Ülkemizde, Çevre ve Şehircilik Bakanlığı portal çalışmaları içerisinde bulunmaktadır. Bu portal çalışmalarında hızlı portal geliştirimini destekleyecek teknolojilere gerek duyulmaktadır. Geliştirilecek konumsal portallar, portlet teknolojisi kullanılarak gerçekleştirilmesi, bu aşamadan sonra UKVA ortamında geliştirilecek portalların hızlı geliştirilmesine büyük destek sağlayacaktır. Konumsal portallar için "konumsal portletler" geliştirilerek, konumsal portallar de geliştirilen web uygulamalarının "yeniden kullanılabilirliği" sağlanabilecektir.

Konumsal portalların günümüzde olduğundan daha fazla işlevi yerine getirebilmesi için bir yaklaşım ortaya konmuş ve bu doğrultuda bir prototip portal geliştirilmiştir. Kullanıcının,

portal üzerinde konumsal analizleri gerçekleştirebildiği bir ortam oluşturulmuştur. Konumsal analizlerinin gerçekleştirilmesi için OGC WPS işlemleri kullanılmıştır. Uygulama odaklı konumsal portal geliştirimi ile portal kullanıcılarının, ihtiyaç duyduğu veriye daha kolay ve hızlı bir şekilde ulaşması sağlanmış, konumsal portal ortamında bu konumsal veriler üzerinde analizler gerçekleştirebildiği bir ortam oluşturulmuştur. Konumsal portal ortamında gerçekleştirilecek uygulamanın bir WPS işlemi tarafından gerçekleştirilemeyeceği durumda ise portal ortamında WSK uygulamaları gerçekleştirilmiştir.

Çalışmanın temel ilgi alanlarından bir diğeri ise, WSK bakımından sunacağı katkı için Portal teknolojisinin değerlendirilmesidir. Bu amaçla, örnek bir WSK senaryoları belirlenmiş, iki genel WSK tarzında gerçekleştirilmiştir. Birinci tarzda, OGC Web Processing Servisi (WPS) ve onun için tanımlanmış WSK yöntemleri denenmiştir. İkinci tarzda, "Portlet Eventing" yoluyla örnek WSK gerçekleştirilmiş ve her iki tarz karşılaştırılmıştır. Sonuçta, Portal Teknolojisinin WSK açısından W3C Web Servisleri Mimarisi (WSM) (W3C, 2004) ile öngörülen WSK tarzına göre bir avantaj sunmadığı tespit edilmiştir.

Portal Teknolojisi, WSK açısından da bu andaki en pragmatik çözümü sunduğu söylenebilir. Her ne kadar bu tarzdaki geliştirilecek uygulamalar, iş mantığı bağlamında "sıkı-bağlı" (tightly coupled) olsa da, bu durum, UKVA gibi, servis sağlayıcıların genel amaçlı Web ortamına kıyasla çok daha kısıtlı olduğu ortamlar açısından çok büyük bir dezavantaj oluşturmayacaktır.

Çalışmada ayrıca, WPS standardında yer alan WSK yöntemleri sınıflandırmasının yeterince kapsayıcı olmadığı görüşünden hareketle, standart için yeni bir sınıflandırma önerilmiştir. Diğer yandan, mevcut WPS sunucuları incelenmiş ve sunuculardaki çeşitli eksikler dile getirilmiş ve çözüm önerisi ortaya konulmuştur.

Bu çalışmada, bir konumsal portalin tamamıyla, açık ve özgür kaynak kodlu yazılımlar ile gerçekleştirilebileceği ispatlanmış ve bu doğrultuda portal bileşenleri, kullanılacak standartlar ve portal mimarisi belirlenmiştir.

Açık ve özgür kodlu yaklaşım, konumsal portal mimarisinde ikinci olarak, UKVA ortamında geliştirilecek konumsal portletler için oluşturulacak portlet kataloğu yaklaşımında kullanılmıştır. UKVA ortamında yapılacak konumsal portal çalışmalarında, geliştirilecek konumsal portletlerin açık ve özgür kaynak kodlu olarak portlet kataloğunda paylaşılması ile UKVA ortamında gerçekleştirilecek diğer konumsal portal çalışmalarındaki portal geliştiricilerinin, portlet kataloğundaki konumsal portletleri kendi ihtiyaçları doğrultusunda geliştirmesiyle, UKVA ortamında hızlı ve sunduğu hizmet bakımından kaliteli konumsal portal üretimleri sağlanabilecektir.

KAYNAKLAR

Abdelnur, A., Hepper, S., 2003. Java Portlet Specification, Version 1.0, Status: FCS Specification, Specification Lead: Sun Microsystems, Inc., Release: August 29.

Akıncı, H., 2006. *Konumsal Veri Altyapılarının Web Servisleri ile Gerçekleştirilmesi: Mevcut Durum Analizi Ve Gelecek Yönelimlerinin Belirlenmesi*, Doktora Tezi, K.T.Ü Fen Bilimleri Enstitüsü, Eylül 2006, Trabzon.

Akram A., Chohan D., Wang X.D., Yang X., Allan R., 2005. A Service Oriented Architecture for Portals Using Portlets. CCLRC e-Science Centre, CCLRC Daresbury Laboratory, Warrington WA4 4AD, UK.

Stollberg B., Zipf A., 2007. OGC Web Processing Service Interface for Web Service Orchestration Aggregating geo-processing services in a bomb threat scenario, 2007.

Stollberg B., Zipf A., 2009. Geoprocessing Services for Spatial Decision Support in the Domain of Housing Market Analyses Experiences from Applying the OGC Web Processing Service Interface in Practice, 2009.

Cömert Ç., 2004. Web Services and National Spatial Data Infrastructures, International Society of Photogrammetry and Remote Sensing, XXth Congress, Commission IV, WG IV/4, Spatial Data Infrastructure, 12-23 July 2004, Istanbul, Turkey.

Hepper, S., 2008. Java Portlet Specification, Version 2.0, Status: FCS Specification, Specification Lead: Sun Microsystems, Inc., Release: January 25.

OASIS, 2007. OASIS Web Services Business Process Execution Language (WSBPEL) TC Committee Specification Version 2.0.

OGC, 2005. OpenGIS Web Processing Service, Version: 0.4.0, OGC 05-007r4, OGC Discussion Paper.

OGC, 2007. OpenGIS Web Processing Service, Version: 1.0.0, OGC 05-007r7, OGC Standard.

W3C, 2004. Web Services Architecture, WC3 Working Group Note, 11 February 2004. <http://www.w3.org/TR/ws-arch/>

URL 1, Sunucuya gönderilen GET isteklerinde karakter sınırlaması, <http://support.microsoft.com/default.aspx?scid=KB;en-us;q208427> (En son Giriş Tarihi: 15.04.2013)

URL 2, Açık ve Özgür Kaynak kod yaklaşımlarının İrdelenmesi 1, <http://www.gnu.org/philosophy/open-source-misses-the-point.tr.html> (En son Giriş Tarihi: 15.04.2013)

URL 3, Açık ve Özgür Kaynak kod yaklaşımlarının İrdelenmesi 2, <http://www.webcitation.org/5TchyyzYm> (En son Giriş Tarihi: 15.04.2013)

URL 4, Açık ve Özgür Kaynak kod yaklaşımlarının İrdelenmesi 3, <http://cs-exhibitions.uni-klu.ac.at/index.php?id=224> (En son Giriş Tarihi: 15.04.2013)