

**KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

HARİTA MÜHENDİSLİĞİ ANABİLİM DALI

**KONUMSAL VERİ ALTYAPILARI İÇİN
KONUMSAL PORTAL GELİŞTİRİM ÇERÇEVESİ**

YÜKSEK LİSANS TEZİ

Harita Müh. Muhammet Emre YILDIRIM

**MAYIS 2012
TRABZON**

**KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

HARİTA MÜHENDİSLİĞİ ANABİLİM DALI

**KONUMSAL VERİ ALTYAPILARI İÇİN
KONUMSAL PORTAL GELİŞTİRİM ÇERÇEVESİ**

Harita Müh. Muhammet Emre YILDIRIM

**Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünde
“HARİTA YÜKSEK MÜHENDİSİ”
Unvanı Verilmesi İçin Kabul Edilen Tezdir.**

**Tezin Enstitüye Verildiği Tarih: 02.05.2012
Tezin Savunma Tarih : 25.05.2012**

Tez Danışmanı: Prof. Dr. Çetin CÖMERT

Trabzon 2012

Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü
Harita Anabilim Dalında
Muhammet Emre YILDIRIM tarafından hazırlanan

KONUMSAL VERİ ALTYAPILARI İÇİN
KONUMSAL PORTAL GELİŞTİRİM ÇERÇEVESİ

başlıklı bu çalışma, Enstitü Yönetim Kurulunun 08.05.2012 gün 1455 sayılı kararıyla
oluşturulan jüri tarafından yapılan sınavda

YÜKSEK LİSANS TEZİ
olarak kabul edilmiştir.

Jüri Üyeleri

Başkan : Prof.Dr. Çetin CÖMERT
Üye : Doç.Dr. Fevzi KARSLI
Üye : Yrd. Doç.Dr. Mustafa ULUTAŞ

Prof.Dr. Sadettin KORKMAZ
Enstitü Müdürü

ÖNSÖZ

“Konumsal Veri Altyapıları için Konumsal Portal Geliştirim Çerçevesi” başlıklı çalışma, Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Harita Mühendisliği Anabilim Dalında yüksek lisans tezi olarak hazırlanmıştır.

Yüksek lisans tezimin danışmanlığını üstlenerek, tez çalışması süresince üst görüşü ve önerileri ile yol gösteren, değerli katkıları ile bana destek olan, ilgi ve yardımını esirgemeyen değerli hocam Sayın Prof. Dr. Çetin CÖMERT’e sonsuz teşekkürlerimi sunarım.

Çalışmaya “Konumsal Veri Altyapıları için Bileşen Geliştirme ve Trabzon İli Örneğinde Uygulama” adlı projeye destek olan Karadeniz Teknik Üniversitesi Rektörlüğüne teşekkür ederim. Tez aşaması boyunca daima desteğini gördüğüm, Yrd.Doç.Dr. Halil AKINCI ve Yrd.Doç.Dr. Hasan Tahsin BOSTANCI teşekkürlerimi sunarım. Tez sürecinde desteğini esirgemeyen sevgili abim bölüm sekreterimiz Muhammet GÜMRÜKÇÜOĞLU’ na sonsuz teşekkürlerimi sunarım.

Ayrıca bugünlere gelmemde, büyük fedakârlıklar gösteren aileme ve dostlarıma sonsuz teşekkürlerimi sunarım. Bu tez çalışmamı, bugünlere gelmemde ve bu çalışmayı yapabilmemdeki en büyük paya sahip olan tez sürecinde kaybettiğim çok kıymetli anneme adıyorum.

Muhammet Emre YILDIRIM

Trabzon 2012

TEZ BEYANNAMESİ

Yüksek Lisans Tezi olarak sunduđum “Konumsal Veri Altyapıları için Konumsal Portal Geliřtirim Çerçevesi” bařlıklı bu çalıřmayı bařtan sona kadar danıřmanım Prof.Dr. Çetin Cömert’in sorumluluđunda tamamladıđımı, verileri/örnekleri kendim topladıđımı, deneyleri/analizleri ilgili labaratuarda yaptıđımı/yaptırdıđımı, bařka kaynaklardan aldıđımı bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiđimi, çalıřma sürecinde bilimsel arařtırma ve etik kurallara uygun olarak davrandıđımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiđimi beyan ederim. 10/05/2012

Muhammet Emre YILDIRIM

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ	III
TEZ BEYANNAMESİ	IV
İÇİNDEKİLER	V
ÖZET	VII
SUMMARY	VIII
ŞEKİLLER DİZİNİ	IX
TABLolar DİZİNİ	XI
SEMBOLLER DİZİNİ	XII
1. GENEL BİLGİLER.....	1
1.1. Giriş	1
1.2. Portal Teknolojisi	4
1.3. Java Portlet Belirtimi	9
1.3.1. Portlet	10
1.3.2. Portlet Container	11
1.3.3. Portal Sayfası	11
1.3.4. Portlet Yaşam Döngüsü	12
1.3.5. Portlet Modları	13
1.3.6. Portlet Pencere Durumları	14
1.4. “Portlet Eventing” mekanizması	16
1.5. Neden Portlet Tabanlı Portal	17
1.6. Web Services for Remote Portlets Belirtimi	19
1.7. Web Processing Service	24
1.7.1. WPS Operasyonları	26
1.7.1.1. GetCapabilities Operasyonu	26
1.7.1.2. DescribeProcess Operasyonu	29
1.7.1.3. Execute Operasyonu	33
1.8. Web Servisleri Tanımlama Dili	37
2. YAPILAN ÇALIŞMALAR	39
2.1. UKVA Konumsal Portal Çerçevesi	39

2.2.	Yeniden Kullanılabilir Portlet Tabanlı Prototip Konumsal Portal Geliştirimi	44
2.3.	Portal Teknolojisi ve Web Servisi Kompozisyonu	52
2.4.	WPS ile Web Tabanlı Konumsal Analiz	54
2.5.	WPS ile WSK	59
2.5.1.	Koşum Zamanında Servis Kompozisyonu	60
2.5.1.1.	KVP Kodlanmış İsteklere Az Sayıya Servisin Kompozisyonu	61
2.5.2.	Tasarım Zamanında Servis Kompozisyonu	62
2.5.2.1	Yeni Bir WPS İşlemi ile WSK	62
2.5.2.2	Bir İş Akışı Motoru ile Servis Kompozisyonu	64
2.6.	Geliştirilen Uygulama Tabanlı Konumsal Portalların İrdelenmesi ve Prototip Konumsal Portal Geliştirimi	67
2.6.1.	AWARE Konumsal Portal Mimarisi	68
2.6.2.	GENESIS Konumsal Portal Mimarisi	71
2.6.3.	AWARE ve GENESIS Konumsal Portallarının İrdelenmesi	73
2.6.4.	Katalog Konumsal Portal ve Uygulama Konumsal Portalını Karşılaştırılması	74
2.6.5.	Uygulama Tabanlı Prototip Konumsal Portal Geliştirimi	77
2.7.	Konumsal Portal Geliştiriminde Açık ve Özgür Kaynak Kodlu Yaklaşım	80
3.	BULGULAR VE İRDELEMELER	86
4.	SONUÇLAR	94
5.	KAYNAKLAR	98
6.	EKLER	104
	ÖZGEÇMİŞ	

Yüksek Lisans

ÖZET

KONUMSAL VERİ ALTYAPILARI İÇİN KONUMSAL PORTAL GELİŞTİRİMİ ÇERÇEVESİ

Muhammet Emre YILDIRIM

Karadeniz Teknik Üniversitesi

Fen Bilimleri Enstitüsü

Harita Mühendisliği Anabilim Dalı

Danışman: Prof.Dr. Çetin CÖMERT

2012, 117 Sayfa, 18 Ek Sayfalar

Bu tez çalışmasında, dünya genelinde UKVA gerçekleştiriminin en popüler yöntemi olan konumsal portalların 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 portalın 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 portalın tamamıyla açık ve özgür kaynak kodlu yazılımlar ile gerçekleştirilebileceği ispatlanmış ve bu doğrultuda portal bileşenleri, kullanılacak standartlar ve portal mimarisi belirlenmiştir.

Anahtar Kelimeler:Konumsal Portal, Servis Yönelimli Mimari, Web Servisleri Kompozisyonu, Web Processing Servisi, Ulusal Konumsal Veri Altyapısı

Master Thesis

SUMMARY

**A FRAMEWORK FOR GEOPORTAL DEVELOPMENT IN
SPATIAL DATA INFRASTRUCTURES**

Muhammet Emre YILDIRIM

Karadeniz Technical University

The Graduate School of Natural and Applied Sciences

Geomatics Engineering Graduate Program

Supervisor: Prof.Dr. Çetin CÖMERT

2012, 118 Pages, 18 Pages Appendix

In this thesis, a framework was defined for geoportals which is the most popular method for the worldwide National Spatial Data Infrastructures (NSDI) activities. This framework includes three components. These are “rapid geoportal development”, “web service composition in Geoportal environments”, “open and free geoportal development”. The rapid geoportal development would enable a rapid development of portal by developers which is valuable especially for countries that have been late in adopting an NSDI. We realized the rapid geoportal development component with geoportals which are developed with standards. Thus, the “re-usability” and “interoperability” problems for geoportals are dealt with. Web Service Composition (WSC) at geoportal environment component is developed to provide more complicated services than search, view and download services for portal users. Portal technology for WSC was studied for in terms of which has not been realized in the way that it should be in the Spatial Data Management Communities. Web Processing Service which is an OGC standard was used for realizing spatial analysis at geoportal environment. Classification of the WSK methods at WPS standard was examined and a new classification is offered for this standard. In this study, current WPS servers were examined as well. Some shortcomings of these servers were identified and solutions for these were offered. With open and free geoportal development component, the idea that a geoportal can be developed completely with open and free software has been proven.

Key Words: Geoportal, Service Oriented Architecture, Web Service Composition, Web Processing Service, National Spatial Data Infrastructure

ŞEKİLLER DİZİNİ

	<u>Sayfa No</u>
Şekil 1. Portal sınıflandırması	6
Şekil 2. Inspire konumsal portalından ekran görüntüsü	7
Şekil 3. GEOSS konumsal portalından ekran görüntüsü	7
Şekil 4. GENESIS konumsal portalından ekran görüntüsü	8
Şekil 5. Portal Sayfasının Bileşenleri	10
Şekil 6. Portal Sayfasının Oluşturulması	11
Şekil 7. Portal sayfası üretim iş akışı diagramı	13
Şekil 8. Örnek portlet	14
Şekil 9. NORMAL pencere durumu	15
Şekil 10. MAXIMIZED pencere durumu	15
Şekil 11. MINIMIZED pencere durumu	15
Şekil 12. WSRP aktörleri.....	20
Şekil 13. WSRP iş akışı	23
Şekil 14. WPS iş akışı	25
Şekil 15. WSDL doküman yapısı	37
Şekil 16. UKVA konumsal portal geliştirimi için ortaya koyulan çerçeve....	40
Şekil 17. UDDI kataloğuna portletlerin kaydedilmesi	41
Şekil 18. Harita görüntüleyici portleti	45
Şekil 19. Arama Portleti	45
Şekil 20. Metadata Editör Portleti	46
Şekil 21. Yeniden kullanılabilir portletlerle geliştirilen prototip portal mimarisi.....	47
Şekil 22. Çalışma kapsamında geliştirilmiş prototip konumsal portaldan ekran görüntüsü	47
Şekil 23. GOS portaldan ekran görüntüsü	49
Şekil 24. GOS portal mimarisi	49
Şekil 25. T.C. Orman ve Su İşleri Bakanlığı bünyesinde gerçekleştirilen konumsal portal uygulaması	51
Şekil 26. Harita görüntüleme arayüzü	52
Şekil 27. Portlet Eventing ile web servisi kompozisyonuna basit düzeyde örnek	53

Şekil 28.	Örnek WSK iş akışı UML diagramı	54
Şekil 29.	Buffer işleminin iş akışı	58
Şekil 30.	Geliştirilen WPS işleminde iş akışı UML diagramı	63
Şekil 31.	Taverna Workbench ile oluşturulan iş akış diagramı	66
Şekil 32.	AWARE konumsal portal mimarisi	69
Şekil 33.	Yağış miktarı ölçen istasyonların dağılımı ve havza sınırını gösteren portal ekranı	70
Şekil 34.	Havza sınırı içerisinde yükseklik bölgelerini gösteren portal ekran görüntüsü	70
Şekil 35.	GENESIS konumsal portal mimarisi	72
Şekil 36.	GENESIS projesi kapsamında kullanılan WPS işlemleri	73
Şekil 37.	Katalog konumsal portal kullanılarak uygulama gerçekleştirmek isteyen kullanıcının işlem adımları	76
Şekil 38.	Çalışma kapsamında geliştirilen konumsal portal mimarisi	79
Şekil 39.	Konumsal portaldan ekran görüntüsü	80
Şekil 40.	Portal mimarisi ve kullanılan açık ve özgür kaynak kodlu yazılımlar	84

TABLolar DİZİNİ

	<u>Sayfa No</u>
Tablo 1. WPS operasyonları ve sağlanması gereken “encoded” yöntemi	26
Tablo 2. GetCapabilities isteği için tanımlanan parametreler	27
Tablo 3. Capabilities dokümanının yapısı	28
Tablo 4. DescribeProcess isteği için tanımlanmış parametreler	30
Tablo 5. DescribeProcess dokümanının yapısı	31
Tablo 6. Execute isteği için tanımlanmış parametreler	33
Tablo 7. Execute isteği sonucunda üretilecek dokümanının yapısı	35
Tablo 8. WPS sunucularının OGC WPS standardına göre uyumluluk testi	56
Tablo 9. Tez çalışması kapsamında geliştirilen WPS işlemleri	57
Tablo 10. AWARE konumsal portalında kullanılan WPS işlemleri	71

SEMBOLLER DİZİNİ

API	: Application Programming Interface
AWARE	: A tool for monitoring and forecasting Available WAtER REsource in mountain environment
BPEL	: Business Process Execution Language
CBS	: Coğrafi Bilgi Sistemleri
CSW	: Catalogue Service Web
EUOSME	: The European Open Source Metadata Editor
FOSS	: Free and Open Source Software
GENESIS	: Generic European Sustainable Information Space for Environment
GEOSS	: Global Earth Observation System of Systems
GEOTIFF	: Geo- Tagged Image File Format
GML	: Geographical Markup Language
GOS	: Geospatial One-Stop
HTTP	: Hypertext Transfer Protocol
INSPIRE	: Infrastructure for Spatial Information in Europe
ISO	: International Organization for Standardization
JCP	: Java Community Process
JSR 286	: Java Portlet Specification 2.0
KVA	: Konumsal Veri Altyapısı
KVP	: Key Value Pairs
OASIS	: Organization for the Advancement of Structured Information Standards
OGC	: Open Geospatial Consortium
PT	: Portal Teknolojisi
SYM	: Servis Yönelimli Mimari

SZ	: Servis Zincirleme
UDDI	: Universal Description, Discovery and Integration
UKVA	: Ulusal Konumsal Veri Altyapısı
XML	: Extensible Markup Language
W3C	: World Wide Web Consortium
WCS	: Web Coverage Service
WFS	: Web Feature Service
WMS	: Web Mapping Service
WPS	: Web Processing Service
WSBPEL	: Web Services Business Process Execution Language
WSDL	: Web Service Description Language
WSK	: Web Servisi Kompozisyonu
WSRP	: Web Services for Remote Portlets

1. GENEL BİLGİLER

1.1. Giriş

Konumsal Veri Altyapıları (KVA), günümüzde önem kazanmış ve gerçekleştirilmesi zorunlu hale gelmiştir. 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. Günümüzde, veri altyapılarının gerçekleştirilmesindeki en önemli bileşenlerden bir tanesi "web servisleri"dir. Verinin değil, servislerin kullanıcılara sunulması, servis sağlayıcılar tarafından sunulan pek çok servisin bir arada kullanılarak, kullanıcıların ihtiyacına göre hizmet sunmanın önemi artmaktadır. Servislerin bir arada kullanılarak kullanıcıya hızlı ekonomik ve kaliteli hizmetin sunulması, ancak Servis Yönelimli Mimari (SYM) ile mümkün olacaktır. SYM de, web servisleri kompozisyonu (WSK) sorununun etkin bir çözüme kavuşturulamadığı görülmüştür. SYM'nin gerçekleştirilmesinde önemli bir role sahip olan portallar, Konumsal Veri Altyapılarının gerçekleştirilmesinde de önemli bir rol oynayacaktır. Portallar ve portletler, etkin bir SYM ortamına geçiş sürecinde ara aşama olarak görülmektedir (Akram vd, 2005). Bu bakımdan, kamu kurumları, yerel yönetimler, özel sektör ve konumsal veri ile iş yapan kesimler arasındaki birlikte işlerliği gerçekleştirmek isteyen ülkeler, konumsal portallarını hızla geliştirmişlerdir.

Konumsal Veri Altyapılarının gerçekleştirilmesinde önemli role sahip olan portalların hızla geliştirilebilmesi önem arz etmektedir. Sadece KVAların gerçekleştirilmesindeki rolü gereği değil, Web2.0 uygulamalarının dünyada önem kazanmasıyla portal teknolojisinde hızlı bir gelişim meydana gelmiştir. Konumsal portalların son kullanıcıya sunduğu, ihtiyaç duyduğu konumsal veri için arama, izleme ve indirme özelliklerinin yanısıra, kişiselleştirme özelliğiyle KVA ların gerçekleştirilmesinde büyük kolaylıklar sağlayacaktır. Örneğin; INSPIRE direktifleri çerçevesinde, pek çok Avrupa birliği üyesi ülkeler, INSPIRE konumsal portalında olduğu gibi kendi konumsal portallarını geliştirmişlerdir.

Bu çalışma kapsamında, ülkelerin kendi UKVA larını gerçekleştirilmesi çalışmaları için geliştirilmiş portalların bir kısmı incelenmiştir¹. Dünya genelinde konumsal portallar incelendiğinde bu portalların büyük bir kısmının katalog portalı sınıfına girdiği görülmektedir. Katalog konumsal portalları genellikle merkezi bir meta veri kataloğuna sahiptirler. Konumsal bilgi kaynağı sağlayıcıları, sahip oldukları bilgi kaynaklarını portal katalogu üzerinden yayımlarlar. İstemciler de portal kataloğunda arama yaparak uygulama gereksinimlerini karşılayan veri ve/veya servisleri bulurlar ve onları ilgili sağlayıcılardan isteyerek uygulamalarını gerçekleştirirler (Akıncı ve Cömert, 2007). Dünya genelinde geliştirilen konumsal portallar incelendiğinde çoğu konumsal portalın, arama, izleme, indirme ve kişiselleştirilme gibi temel düzeydeki servisleri kullanıcıya sunduğu görülmektedir. Örneğin; INSPIRE kapsamında geliştirilen Inspire Konumsal Portalında, “Metada Editor”, “Viewer”, “Metadata Validator”, “Discovery” gibi servisler sunulmaktadır. Kullanıcı bu portalda ihtiyaç duyduğu konumsal veri, veriseti veya konumsal servisleri için “Discovery” bölümünden arama yapabilmekte ve “Viewer” servisiyle bulduğu sonuçları görüntüleyebilmektedir. Konumsal veriseti, veriseti serileri ve konumsal servislerini yayınlamak isteyen kullanıcılar için “Metadata Editor” ve “Metadata Valitador” servisleri sunulmaktadır.

Konumsal portallarda 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 portallar için benzer veya aynı uygulamalar tekrar tekrar geliştirilmiştir. Çünkü geliştirilen web uygulamaları yeniden kullanılabilir özellikte değildir. Türkiye gibi dünyadaki UKVA çalışmalarından geri kalan ülkeler, konumsal portallarını kısa sürede gerçekleştirmeye ihtiyaç duymaktadırlar. Dolayısıyla hızlı portal geliştiriminin sağlanması için geliştirilecek uygulamaların yeniden kullanılabilir özellikte ve kolayca portal çalışma ortamına entegre edilebilir olması gerekmektedir.

¹INSPIRE Geoportal <http://www.inspire-geoportal.eu/>
Geoportail France, <http://www.geoportail.fr/>
IDEE portal, http://www.ideo.es/show.do?to=pideep_pidee.ES
Go-Geo! Portal, <http://www.gogeo.ac.uk/cgi-bin/index.cgi>
Portale Cartografico Nazionale, <http://www.pcn.minambiente.it/PCN/>
Geospatial Data Service Centre Portal , <http://gdsc.nlr.nl/gdsc/en>
GNII geoportal, <http://www.geoportal.it/>
Geo.admin.ch <http://www.geo.admin.ch/internet/geoportal/en/home.html>

Bu çalışmada amaç, Ulusal Konumsal Veri Altyapıları (UKVA) nın gerçekleştirilmelerinde, halen önemli rol üstlenmiş olan geoportallerin geliştirilmesinde Portal Teknolojisi (PT) nin sunduğu olanakların belirlenmesidir. Bu bakımdan PT'nin sunabileceği olanaklardan biri "portlet yeniden kullanılabilirliği" dir. UKVA'nın gerçekleştirilmesi sürecinde geliştirilecek konumsal portalların hızla geliştirilebilmesi bakımından önemlidir. 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) standartlarına uygun portletlerdir. PT' nin SYM de hala tatmin edici bir çözüme kavuşturulamamış olan Web Servisleri Kompozisyonu (WSK) bakımından sunabileceği katkının belirlenmesi, bu çalışmanın diğer amacıdır.

Bu çalışma kapsamında ayrıca, bir literatür taraması yapılarak, portal teknolojisi alanındaki son durum ve portal teknolojisinin Servis Yönelimli Mimari (SYM) açısından nasıl bir katkı sunabileceği belirlenmeye çalışılmıştır. Bu kaynaklardan bazıları (Akram vd, 2005), (Rao, 2009), (North, 2006), (Maguire vd, 2005), (Diaz vd, 2004) ve (Yang, 2007) dir. Bu inceleme sonucunda oluşan görüşümüze göre, portal teknolojisi alanında bugün en önemli sorunun portlet birlikte işlerliğidir. Çözüme yönelik çeşitli öneriler mevcuttur. Bunlardan biri Diaz vd. (2005) nin önerisidir. Portlet birlikte işlerliği hem portletlerin yeniden kullanılabilirliği ve dolayısıyla hızlı portal geliştirimi ve hem de SYM de hala tatmin edici bir çözüme kavuşturulamamış olan servis kompozisyonu ya da servis zincirleme bakımından çok önemli bir konudur.

Dünya genelinde KVA çalışmaları incelendiğinde, KVA çalışmalarında önde olan ülkelerin konumsal portal geliştirimini tamamladıkları ve süreç içerisinde sürekli geliştirmeler yaptıkları gözlemlenmektedir. Geliştirilen portallar incelendiğinde önemli bir kısım, merkezi metaveri kataloğuna sahip katalog portalları olduğu gözlemlenmektedir. Fakat, bu yapı günümüzün ihtiyaçlarını karşılayamamaktadır. Dolayısıyla, katalog portallarından, uygulama tabanlı portallara geçiş yapılması artık zorunlu hale gelmiştir. Uygulama tabanlı portaldan 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 analizlerin web servisleri tarafından gerçekleştirilmesi için, The Open Geospatial Consortium (OGC) tarafından, 2005 yılında Web Processing Service v0.4.0 standardı yayınlandı (OGC, 2005) ve 2007 yılında güncellenerek Web Processing Service v1.0.0 (WPS) standardı yayınlanmıştır (OGC, 2007). Bu servis sayesinde web üzerinden herhangi bir CBS konumsal analiz fonksiyonu kullanılabilir ya da aynı bölgeye ait iki raster görüntüyü karşılaştırılıp değişim analizini gerçekleştirilebilir ya da vektör veri üzerinde genelleştirme işlemleri uygulanabilir. WPS servislerinin kompozisyonu ile, katı atık yer seçimi gibi kompleks bir CBS analizi gerçekleştirilebilmektedir. Bu çalışma kapsamında, WPS standardını sağlayan açık kaynak kodlu yazılımlar incelenmiştir. Bunlar; 52 North WPS, Deegree WPS, Geoserver WPS, pyWPS'dir.

PT'nin WSK bakımından sunacağı olanaklardan biri olan "Portlet Eventing" mekanizmasını kullanarak portletlerin birbirlerine bağlanabilmesidir. 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. Geliştirilen WPS işlemlerinin kompozisyonuyla, portal ortamında katı atık deponi alanı yer seçimi gibi, kompleks bir CBS analizi gerçekleştirilmiştir.

1.2. Portal Teknolojisi

Dünya genelinde ülkelerin KVA çalışmaları incelendiğinde, KVA gerçekleştiriminde kullanılan en popüler yöntem, konumsal portallar olduğu görülmektedir. KVA

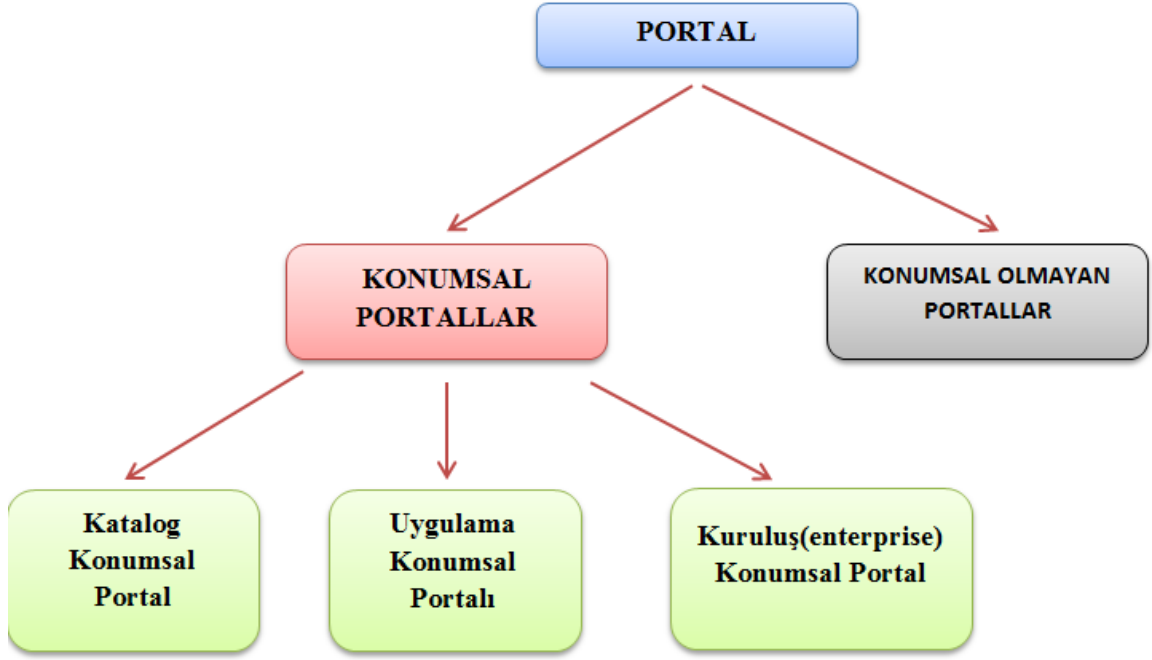
çalışmalarında önde olan ülkeler, konumsal portal geliştirimini tamamlamış ve süreç içerisinde geliştirmeleri yapmaya devam etmektedirler.

Portalın literatürde kabul görmüş bir tanımı bulunmamaktadır. Java Portlet Belirtiminde portal, 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). İçeriklerin birleştirilmesinden kasıt, farklı kaynaklardan elde edilen bilgilerin aynı web sayfasında gösterilmesidir.

INSPIRE kapsamında konumsal portal, INSPIRE direktifleri çerçevesinde hazırlanan AB birliği üye ülkelerin konumsal veri setlerine, konumsal veri ve 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)’ye göre konumsal portal, veri setleri ve servisler içeren bir çevrimiçi konumsal bilgi kaynakları topluluğu için bir kullanıcı arayüzüdür (OGC, 2004).

Literatürde portal sınıflandırması konumsal olan ve konumsal olmayan portallar olarak ikiye ayrılmaktadır. Konumsal portallar kendi içerisinde, metadata portal, uygulama tabanlı portallar ve kurumsal (enterprise) portallar olarak üçe ayrılmaktadır (Foust vd., 2005).



Şekil 1. Portal sınıflandırması

Katalog konumsal portalları genellikle merkezi bir meta veri kataloğuna sahiptirler. Konumsal bilgi kaynağı sağlayıcıları, sahip oldukları bilgi kaynaklarını portal katalogu üzerinden yayımlarlar. İstemciler de portal kataloğunda arama yaparak uygulama gereksinimlerini karşılayan veri ve/veya servisleri bulurlar ve onları ilgili sağlayıcılardan isteyerek uygulamalarını gerçekleştirirler (Akıncı ve Cömert, 2007).

INSPIRE kapsamında geliştirilen konumsal portal da aynı mimariye sahiptir. Kullanıcı, INSPIRE konumsal portalında, “Metadata Editor”, “Viewer”, “Metadata Validator”, “Discovery” gibi servisler sunulmaktadır. Kullanıcı bu portalda ihtiyaç duyduğu konumsal veri, veriseti veya konumsal servisleri için “Discovery” bölümünden arama yapabilmekte ve “Viewer” servisiyle bulduğu sonuçları görüntüleyebilmektedir. Konumsal veriseti, veriseti serileri ve konumsal servislerini yayınlamak isteyen kullanıcılar için “Metadata Editor” ve “Metadata Valitador” servisleri sunulmaktadır. Metadata Editor ile sunulacak konumsal veri hakkında metaveriler oluşturulmaktadır. Metadata Validator ise oluşturulan metaverinin, standartlara uygun üretilip üretilmediğini kontrol etmektedir.

European Commission
INSPIRE Geoportal

EUROPEAN COMMISSION > INSPIRE GEOPORTAL >

Welcome to INSPIRE geoportal

The INSPIRE geoportal provide the means to search for spatial data sets and spatial data services, and subject to access restrictions, view and download spatial data sets from the EU Member States within the framework of the Infrastructure for Spatial Information in the European Community (INSPIRE) Directive. INSPIRE aims at making available relevant, harmonised and quality geographic information to support formulation, implementation, monitoring and evaluation of policies and activities which have a direct or indirect impact on the environment.

What does it include

This version is a prototype INSPIRE geoportal and allows for *discovery* and *viewing* of spatial data sets and services. It's aim is to identify issues related to its implementation and accessing distributed INSPIRE services, to help towards the development of the operational geoportal.

The prototype INSPIRE geoportal currently accesses a limited number of discovery and view services and therefore only a few metadata for spatial data sets and services may be found and viewed. These will increase as more services become available from the EU Member States.

To ensure that the INSPIRE geoportal has a full functional access to the Discovery and View services testing is done in close cooperation with the Member States. This allows performing any necessary modifications or adjustments to improve interoperability and the overall architecture performance and in general facilitate the way towards the full operating capability.

We depend on your feedback to help us make the geoportal better. If you have questions or comments about the site, please [contact us](#).

What's inside

Discovery
Search, discover and access geographic information provided by European governmental, commercial, and non-commercial organizations.
[more...](#)

Viewer
Search, view, edit online maps, and create your personalised map using distributed data.
[more...](#)

Metadata Editor
Create metadata according to the INSPIRE implementing rules.
[more...](#)

NEWS **EVENTS**

Apr 2011

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

NEWS

- 14-Apr-11 INSPIRE Spatial Data Services Recommendations Survey
- 08-Apr-11 Call for participants for data specifications testing of INSPIRE Annex II and III
- 30-Mar-11 Detailed Implementing Rule Development Timetable for Annex II and III
- 30-Mar-11 Availability of updated Technical Guidance documents for INSPIRE Discovery and View Services
- 01-Mar-11 Call for Facilitators for Thematic Working Groups
- 08-Feb-11 INSPIRE: State of Play Reports 2010 published

Şekil 2. Inspire konumsal portalından ekran görüntüsü

Login Register Help About Feedback

HOME BROWSE GALLERY SEARCH GEOVIEWER

Search

View the Gallery
See maps and apps from people and organizations worldwide.

Collaborate
Work together with other users of the Esri Geoportal Server.

Make a Map
Create a map you can view on the web, desktop and mobile devices.

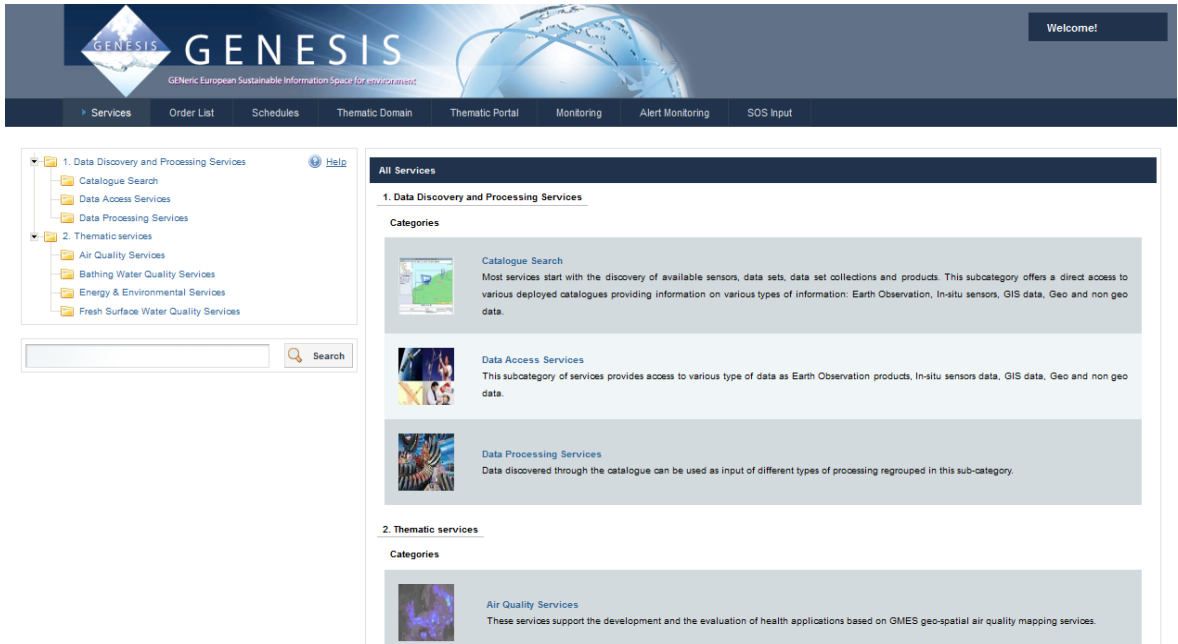
Start ArcGIS Explorer Online
Use a rich application for working with ArcGIS online maps.

Learn More
Esri Geoportal Server is an open source, standards-based solution for discovering and accessing geospatial resources in heterogeneous environments.

esri

Şekil 3. GEOSS konumsal portalından ekran görüntüsü

Uygulama tabanlı portallara örnek gösterilebilecek uygulamalar literatürde mevcuttur. GENESIS ve AWARE konumsal portalları bu konuda örnek gösterilebilir. Kullanıcılar, portal üzerinden çeşitli bölgelere ait hava ve su kalitesi için analizleri konumsal portal üzerinden gerçekleştirebilmektedir. Uygulama tabanlı portalların, metadata portallara göre avantajları bulunmaktadır. Kullanıcı, metadata portalından ihtiyaç duyduğu veriyi metaveriler üzerinden arama yaptıktan sonra, bulduğu sonuçların metaverilerini incelemekte ve ihtiyaç duyduğu veriyi bulduktan sonra, uygulamayı gerçekleştireceği ortama indirmekte, masaüstü CBS yazılımları ile uygulamayı gerçekleştirmektedir. Çoğu zaman, kullanıcı ihtiyaç duyduğu konumsal veriye ulaşmak için saatlerini harcamakta ya da veriye ulaşma mümkün olmamaktadır. Uygulama tabanlı konumsal portalda ise kullanıcı ihtiyaç duyduğu uygulamayı portal üzerinden gerçekleştirmektedir. Kullanıcıya, zaman ve kaynak bakımından kazanç sağlamaktadır. Konumsal analizlerin portal üzerinden gerçekleştirilmesiyle, ek CBS yazılımlarına ihtiyaç duyulmadan konumsal analizler web servisleri tarafından gerçekleştirilebilmektedir. Etkin SYM ortamına geçişte önemli bir aşama bu portallar tarafından gerçekleştirilmiş olacaktır.



Şekil 4. GENESIS Portal Ekran Görüntüsü

Konumsal veri ile uğraşan kesimler arasında birlikte işlerliği sağlayacak konumsal veri altyapılarının oluşturulmasında konumsal portalların rolü büyük olacaktır. Birlikte işlerliğin tam anlamıyla gerçekleştirilmesi ancak SYM ile mümkün olacaktır. Etkin SYM

ortamına geiş srecinde konumsal portallar nemli rol stlenecektir. Dnya genelinde, konumsal portallar incelendiėinde, KVA'lar konumsal portallar ile gerekleřtirmi yoluna gidilmiřtir.

Dnya genelinde, konumsal portallar iin geliřtirilen uygulamaların yeniden kullanılabilir olmadıėı grlmektedir. Bu sorun, her portal uygulamasında aynı ya da benzer uygulamaların tekrar tekrar geliřtirilmesine neden olmuřtur. Bu zaman ve kaynak kaybına yol amaktadır. Bu tez kapsamında konumsal portallar iin dile getirilen yeni yaklařımlardan bir tanesi; portal geliřtiriminin belli standartlara gre yapılması ve portal uygulamalarının yeniden kullanılabilirliėinin saėlanmasıdır. Portal geliřtiriminde kullanılacak standartlar JSR-286 ve WSRP standartlarıdır. Bu standartlar, portal uygulamalarının yeniden kullanılabilirliėini ve portalların birlikte iřlerliėini saėlayacaktır.

1.3. Java Portlet Belirtimi (JSR-286)

Portal teknolojisinin nem kazanmasıyla, Bea, IBM, Oracle gibi bir ok řirket, portlet teknolojisiyle kendi portal bileřenlerini ve rnlerini geliřtirmişlerdir. Her řirket kendi portal zm iin API (Application Programming Interface) yayınlamıřtır. Bu yaklařım, “yeni den kullanılabilirlik ve birlikte iřlerlik” sorununu ortaya ıkarmıřtır. Bir portal ortamında geliřtirilen bir uygulama, farklı bir portal ortamında kullanılamamaktaydı. Dolayısıyla, geliřtiriciler bir ortamda geliřtirdikleri bir uygulamayı bařka bir ortamda tekrar geliřtirmek zorunda kalmıřtır. Bu da zaman ve kaynak kaybına yol amıřtır.

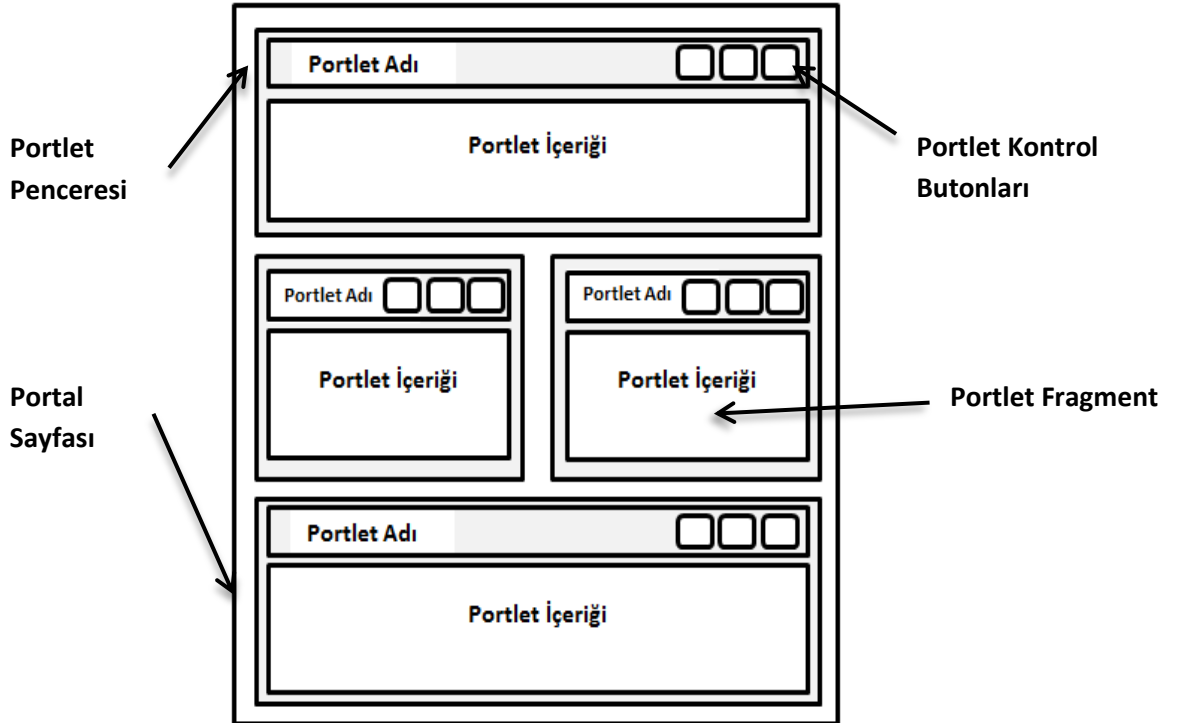
“The Java Community Process (JCP)”, 2003 yılında farklı portal ve portletler arasındaki “yeni den kullanılabilirlik” sorununu zmek iin Java Portlet Belirtimini (JSR 168) yayınlamıřtır (Abdelnur vd., 2003). Portlet geliřtirimi belli standartlara gre yapılması oluřturulan portletlerin yeni den kullanılabilmesini ve standartları saėlayan herhangi bir portal sunucusunda alıřmasını saėlamaktadır. JSR 286 standardı, portletler iin “Portlet API v2.0” tanımlayarak, portletler ve portallar arasındaki “yeni den kullanılabilirliėi” saėlamayı amalamaktadır.

JSR 286 standardını sağlayan bütün portletler bu standartları sağlayan tüm portal sunucularda çalışabilmektedir. Bu sayede, Servis Yönelimli Mimaride olduğu gibi farklı platformlarda çalışabilme imkânı, portal mimarisinde kullanıcılara sunulabilmektedir.

Bu standarda göre portal; portletler, portlet container, portal sunucusundan oluşmaktadır. Portal sunucusu ve portlet container tek bir bileşen olarak geliştirilebilir.

1.3.1. Portlet

Portlet, bir portalde sunulmak üzere, bilgi ya da servis niteliğinde belirli bir içerik sunan bir uygulamadır. Portletler, portallar tarafından, bilgi sistemlerine sunum katmanı sağlayan takılıp çıkarılabilen kullanıcı arayüzü olarak, kullanılmaktadır. Portletler tarafından üretilen içerik “fragment” olarak ta adlandırılmaktadır (Hepper, 2008). Yaşam döngüsü Portlet Container tarafından yönetilmektedir. JSR 286 standardını sağlayan bütün portletler bu standartları sağlayan tüm portal sunucularda çalışabilmektedir. Bu sayede, Servis Yönelimli Mimaride olduğu gibi “farklı platformlar”da çalışabilme imkânı, portal mimarisinde kullanıcılara sunulabilmektedir.



Şekil 5. Portal Sayfasının Bileşenleri ((Hepper 2008) ten uyarlanmıştır)

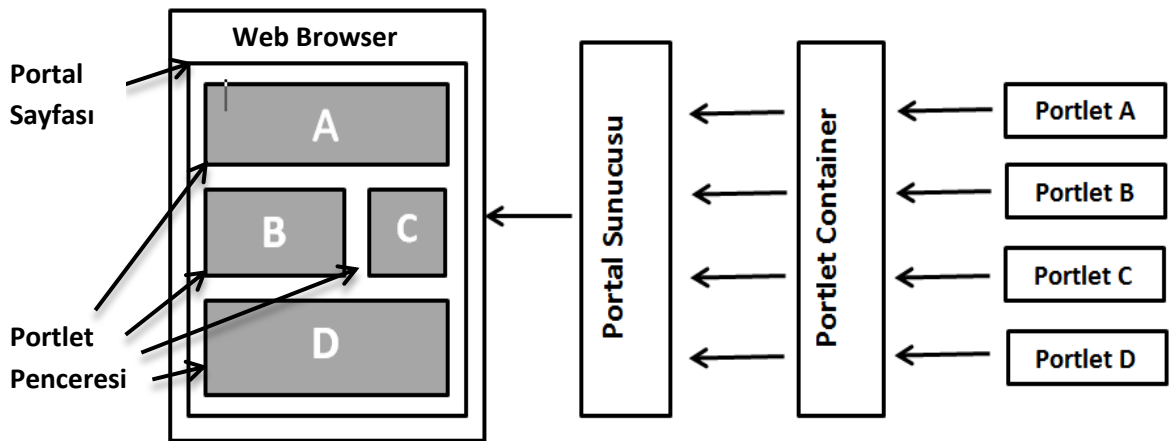
Portletler, portal sayfasında gösterilirken portlet penceresi içerisinde gösterilmektedir. Portlet penceresi, Portlet Adı, portlet kontrol butonları ve portlet tarafından üretilen içerikten oluşmaktadır (Şekil 5).

1.3.2 Portlet Container

Portlet Container, portletlerin yaşam döngüsünü yönetmekte ve portletler için koşum zamanı ortamı (runtime environment) sağlamaktadır. Portlet Container, portal üzerinden kullanıcıların yaptığı istekleri alır, portlete bu isteği iletir ve bu istek sonucunda portletlerin ürettikleri içerikleri portal sunucusuna göndermekle görevlidir. Portletlerin ürettikleri içeriklerin birleştirilip portal sayfasının oluşturulması portal sunucusunun görevidir.

1.3.3. Portal Sayfası

Portal Sayfası, portletlerin ürettikleri içeriklerin gösterildiği portlet pencerelerinden oluşmaktadır. Portal sunucusu da, portletler tarafından üretilen içeriği derleyerek portal sayfası üretir ve kullanıcın web tarayıcısına gönderir. Portal sayfasında birden fazla portlet bulunabilmektedir. Portal sayfasını, kolonlara ve sütunlara bölerek sayfa yapısını ayarlanabilmekte ve portletler bu sayfa yapısına göre yerleştirilmektedir.



Şekil 6. Portal Sayfasının Oluşturulması ((Hepper 2008) ten uyarlanmıştır)

Portal sayfası oluşturulmasının işlem adımları şu şekildedir. Kullanıcı, web tarayıcısı (Internet Explorer, Mozilla Firefox vb.) ya da akıllı mobil cihazlardan, portal sayfasına

bağlanır. Kullanıcı portal üzerinden istekte bulunur. Portal, kullanıcının isteğinin gerçekleştirilmesi için ihtiyaç duyulan portletleri belirler. Portlet container vasıtasıyla istek, portletlere iletilir. İstek sonucu üretilen portlet içeriği (fragment) derlenerek portal sayfası oluşturulur. Oluşturulan portal sayfası, kullanıcının web tarayıcısına gönderilir.

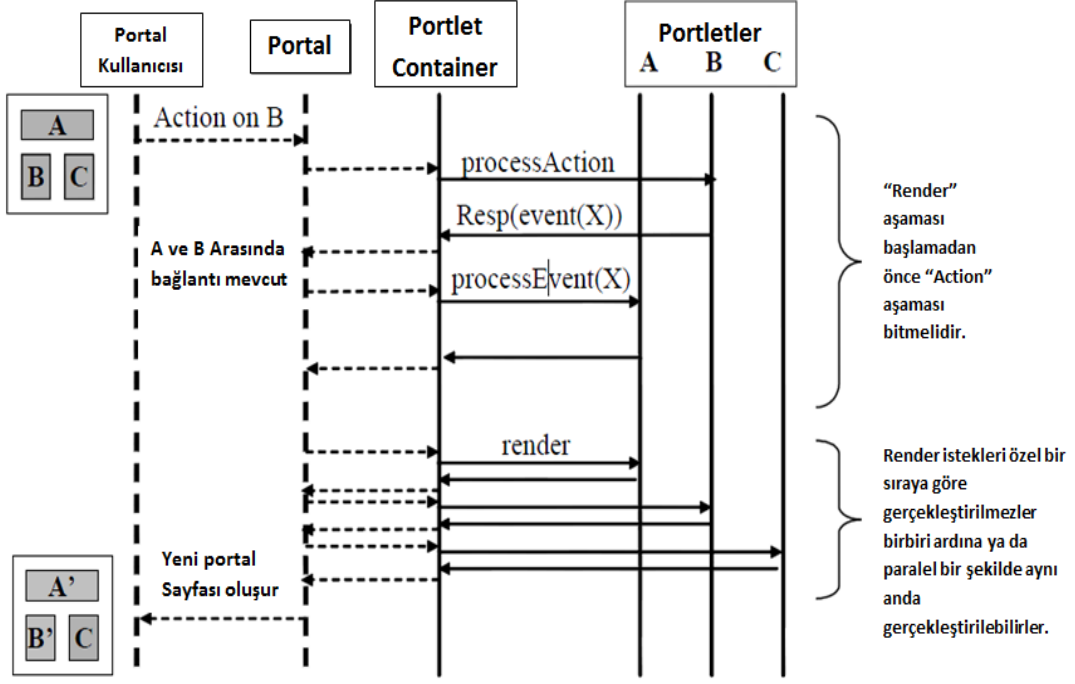
1.3.4. Portlet Yaşam Döngüsü

JS2-286 standardı tarafından tanımlanan portlet arayüzünde, yaşam döngüsünü (life cycle) sağlayan metodlar; “init”, “processAction”, “render” ve “destroy” aşamalarından oluşmaktadır. Portletin yaşam döngüsünü üç aşamada tanımlanmaktadır (Linwood ve Minter, 2004). Bunlar; portletin oluşturulması, portlet isteklerinin işlenmesi ve portletin yaptığı işlemlerin sonlandırılarak portletin portal sayfasında gösterilmesidir.

Portlet Container, portletlerin yaşam döngüsünü sağlamakla yükümlüdür. Portlet uygulaması, Portlet Container başlatıldığında portal sunucusuna yüklenebilir ya da kullanıcının portal üzerinden yaptığı bir istek sonucunda, portlet sunucuya yüklenebilir. Bu işlem için “init” metodu kullanılmaktadır.

Kullanıcı tarafından portal sayfasından yapılan isteklerin işlenmesi aşamasında “processAction” ve “render” metodu kullanılmaktadır. Portlet üzerinden yapılan herhangi bir istek (portlet arayüzünde bir linke veya butona tıklanması vb.) sonucunda o işlemin gerçekleştirilmesi için “processAction” metodu kullanılmaktadır. İşlem sonrasında, portlet arayüzünde gösterilecek içeriğin oluşturulmasında ise “render” metodu kullanılmaktadır.

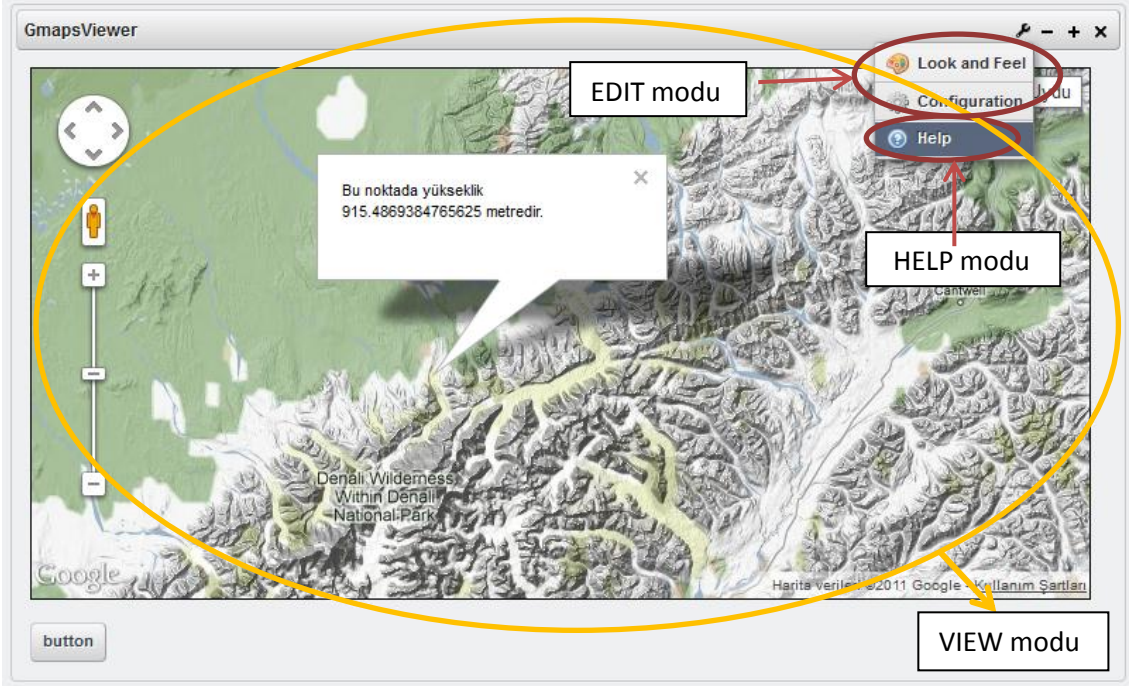
Portletin yaşam döngüsündeki aşamalara ek olarak, “processEvent” aşaması eklenebilmektedir. Bu aşama, portletler arasında iletişim için kullanılan “Events” için kullanılmaktadır. Portletler arasındaki iletişimi sağlayan mekanizmalardan bir tanesi olan “Portlet Eventing”, ilerleyen bölümlerde ayrıntılı irdelenecektir.



Şekil 7. Şekil. Portal sayfası üretim iş akış diagramı

1.3.5. Portlet Modları

JSR-286 standardında, uygulanması zorunlu VIEW, EDIT ve HELP olmak üzere üç mod tanımlanmaktadır. Portlet geliştiricileri bu modlara ek olarak modlar tanımlayabilmektedir. VIEW modu, kullanıcının portal sayfasında portlet içeriğini bulunduran, kullanıcının portlet üzerinde isteklerde bulunabildiği moddur. EDIT modu, portlet uygulaması üzerinde ayarlamalar (configuration) yapabildiği moddur. HELP modu ise portlet uygulaması hakkında kullanıcılara bilgi sunulan moddur. Bu modlar sırasıyla `doView()`, `doEdit()`, `doHelp()` metodlarını kullanmaktadır. Portlet modlarının örnek bir uygulama üzerinde gösterilebilmesi kapsamında temel düzeyde bir portlet geliştirilmiştir.

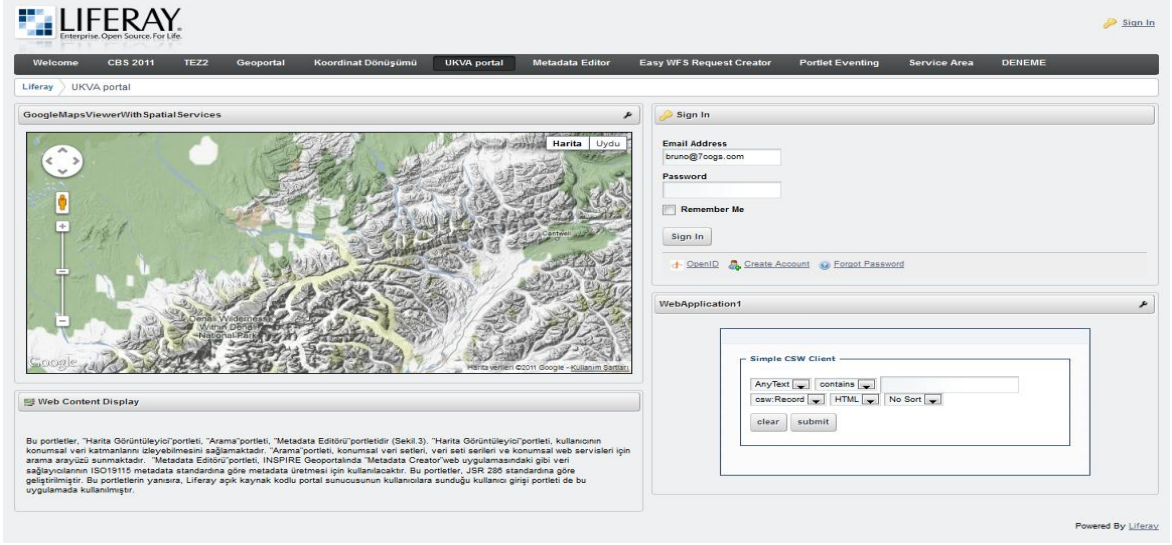


Şekil 8. Örnek Portlet

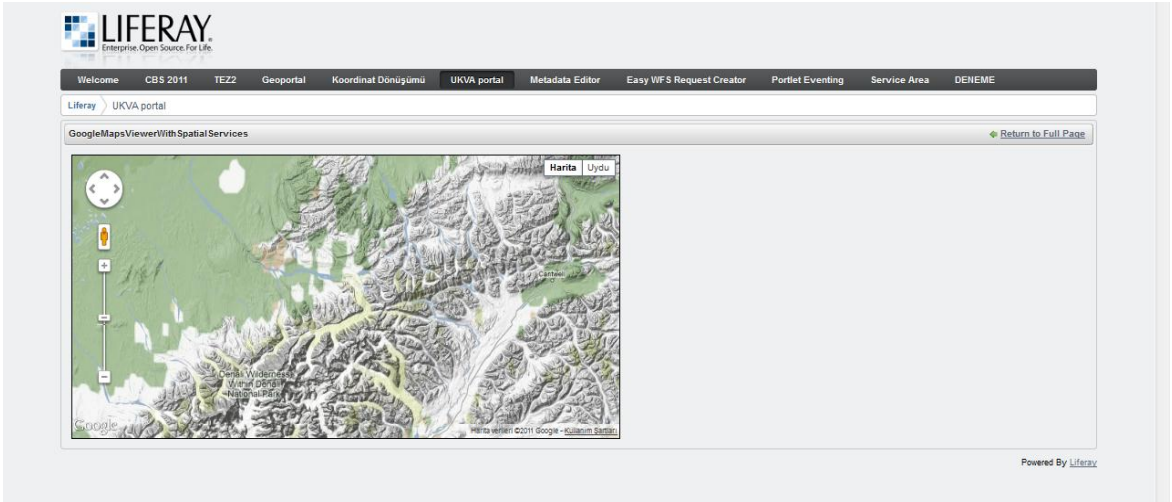
Şekil 8’de görülen örnek portlette VIEW modu aktiftir. EDIT modunu aktif hale getirmek için Şekil8’de görüldüğü gibi “Look and Feel” ve “Configuration” butonuna tıklanır. HELP modunu aktif hale getirmek için “Help” butonuna tıklanır. EDIT modu ile kullanıcıya portlet üzerinde ayarlamalar yapma imkanı verilmektedir. HELP modu ile portlet uygulaması ile neler yapılabileceğine dair yardım bilgisi verilmektedir. Ek 1. de bir portletin modlarının nasıl tanımlanacağına dair örnek bir program dosyası mevcuttur.

1.3.6. Portlet Pencere Durumları (Window States)

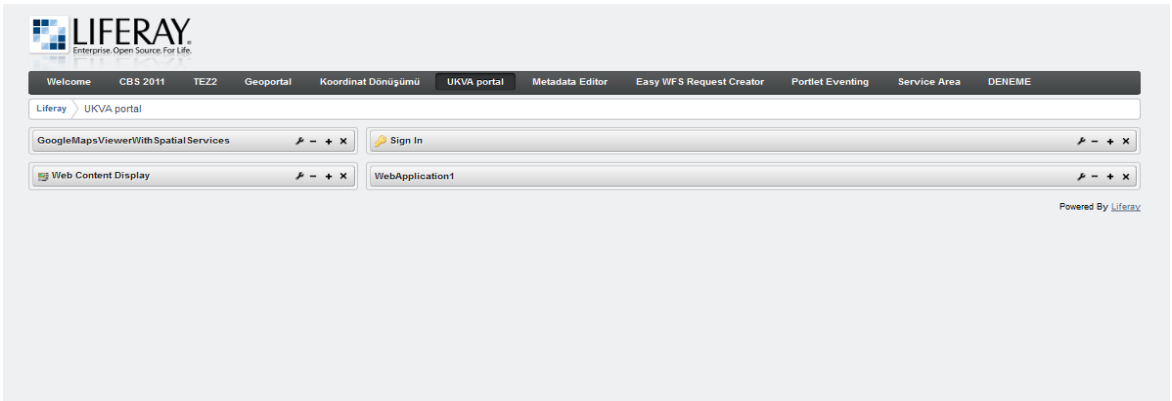
Portlet pencere durumu, portlet tarafından üretilen içeriğin portal sayfasında kaplayacağı yeri belirtmek için kullanılır. Portlet Container, portletin pencere durumu parametresini sunmaktadır. Portal sunucusu bu parametreye göre portletin portal sayfasında gösterimini gerçekleştirmektedir. JSR-286 standardı, portletler için NORMAL, MAXIMIZED and MINIMIZED olmak üzere üç tane pencere durumu tanımlamıştır. NORMAL pencere durumu, bir portal sayfasında birden fazla portletin gösterilebildiği durumdur. MAXIMIZED pencere durumu, bir portal sayfasında sadece bir portletin gösterildiği durumdur. MINIMIZED pencere durumu ise, portletin hiçbir içerik sunmadığı durumdur.



Şekil 9. NORMAL pencere durumu



Şekil 10. MAXIMIZED pencere durumu



Şekil 11. MINIMIZED pencere durumu

1.4.“Portlet Eventing” Mekanizması

Birbirinden bağımsız geliştirilmiş portletlerin birbirleri arasındaki koordinasyonun sağlanabilmesi için JSR 286 standardında, “Portlet Eventing” mekanizması tanımlanmıştır. “Portlet Eventing” portletlerin, portal üzerinde herhangi bir işlem ya da durum değişikliğinde (state changes) portletlerin bu değişikliğe tepki verebilmesi için tasarlanmıştır. “Portlet Eventing”, portlet container tarafından üretilmiş olabilir ya da kullanıcı tarafından bir başka portlet de yapılan bir işlem sonucunda, oluşabilmektedir. Kullanıcının bir portlet üzerinde yaptığı bir işlem sonucu o portlet bir “event” yayınlar ve başka bir portlet bu “event”i işleyerek ve başka bir “event” yayınlayabilir başka bir portlet yayınlanan “event”i kullanarak işlem gerçekleştirebilir ve bu şekilde portal ortamında, portletler birbirleriyle bağlanabilmektedir (Hepper, 2008).

Birbirine bağlanmış iki portletin portal ortamındaki iş akışı; kullanıcı portlet arayüzünde, bir işlem gerçekleştirir ve bunun sonucunda kullanıcının yaptığı istek işlenir (processAction aşaması) ve bu işlem sonucunda portlet event yayınlar (Resp(event(X) aşaması). Daha sonra yayınlanan bu “event”i diğer bir portlet işler (processEvent(X) aşaması). Bundan sonraki aşama Render aşamasıdır. Portal sayfasındaki portletlerin içerikleri, kullanıcının gerçekleştirdiği işlem sonucunda, yeniden üretilerek portal sayfasının yeni hali, kullanıcının tarayıcısına gönderilir.

Bir portlet bir “event” yayınlayabilmesi için “deployment descriptor” dosyası portlet.xml dosyası içinde portlet-application kısmında “event-definition” elementi ile tanımlanması gerekmektedir. Event yayınlamak isteyen portletler, Portlet.xml dosyasında portlet-application kısmında supported-publishing-event elementi ile yayınlamak istediği event tanımlanmalıdır. Aynı şekilde bir başka portlet tarafından yayınlanan “event”i kullanmak için Portlet.xml dosyasında portlet-application kısmında supported-processing-event elementi ile bir başka portlet tarafından yayınlanan “event”i tanımlamak gerekmektedir.

Portlet.xml dosyasında portlet-application kısmında tanımlanmış olan event-definition elementindeki event name, event yayınlayan ve yayınlanan event kullanılır.

portletlerin, supported-publishing-event ve supported-processing-event elementlerinde referanslandırılmalıdır.

Portletler, “event”leri action processing aşamasında setEvent() metodu kullanarak oluşturur. Bu eventler Portlet Container tarafından action processing aşaması bittikten sonra işleyecektir. Portletlerin yayımlanan “event”leri alması için javax.Portlet.EventPortlet arayüzünü sağlaması (implement) gerekmektedir. Portlet Container tarafından processEvent() metodu, eventleri kullanacak tüm portletler için EventRequest ve EventResponse objesi ile birlikte kullanılacaktır. Ek 2’ de bir Portlet Eventing uygulamasına yönelik örnek kodlar mevcuttur.

Portlet.xml dosyasında eventlerin tanımlanması:

```
<portlet-app xmlns="http://java.sun.com/xml/ns/portlet/
  portlet-app_2_0.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/portlet/
  portlet-app_2_0.xsd
  http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd"
  id="myPortletApp" version="2.0">
  <portlet>
  ...
  ...
  </portlet>
  <event-definition>
  <qname xmlns:x="http://sun.com/mapevents" >x:deneme</qname>
  <value-type>
  com.sun.portal.portlet.mapevent.deneme
  </value-type>
  </event-definition>
</portlet-app>
```

Event ismi, qname olarak ve ayırt edilmesi için unique olarak tanımlanır.

1.5. Neden Portlet Tabanlı Portal?

Web 2.0 teknolojilerinin önem kazanmasıyla, portal teknolojisi önem kazanmış, portal ürünleri hızlı bir şekilde geliştirilmiştir. Farklı firmalar tarafından geliştirilen portal ürünleri, portal uygulamaları arasındaki birlikte işlerlik sorunu ortaya çıkmıştır. Bu yüzden, portal uygulamalarının geliştirilmesinde bir standart ihtiyacı ortaya çıkmıştır. Bu ihtiyacı karşılamak için JSR-186 standardı yayınlanmış ve 2008 yılında güncellenerek

JSR-286 standardı son halini almıştır. Bu standartta, portlet mimarisi önerilmiştir. Portal teknolojisi alanındaki firmalar, ürünlerini bu standarda göre geliştirmişlerdir. Böylece farklı portal uygulamaları arasındaki yeniden kullanılabilirlik ve birlikte işlerlik problemine çözüm bulunmuştur. Portlet tabanlı portal geliştiriminin faydaları, 4 ana başlık altında sınıflandırılabilir. Bunlar; Portal Customisation, Standartlara Uyumlu Portal Geliştirimi, Farklı Servislerin Bir Portal Sayfasında Birlikte Çalışabilmesi, Kişiselleştirme, Portletlerin Arasındaki İletişimdir (Osmond ve Guo, 2005).

- Portal Customisation: günümüzde pek çok kurum ve özel şirketler portal uygulamaları kullanmaktadır. Kurumların ve şirketlerin portal ortamındaki ihtiyaçları birbirine benzer ve yakındır. Bir kurum için geliştirilmiş uygulamanın, bir başka kurum tarafından benzeri bir uygulamaya ihtiyaç duyulduğunda, o uygulamanın tekrar baştan geliştirilmeden portal uygulaması üzerinde ihtiyaca göre değişiklik yapılabilmesi zaman ve kaynak kaybını önleyecektir. Portlet tabanlı portal gerçekleştirimi, portal uygulamasını tamamıyla baştan geliştirmek yerine ihtiyaç duyulan değişikliklerin yapılabildiği ortamı portal geliştiricilere sunmaktadır.

- Standartlara Uyumlu Portal Geliştirimi: portal uygulamalarının portlet standartlara göre geliştirilmesi, standardı sağlayan tüm portal sunucularında çalışabilmesini sağlamaktadır. Örneğin, JSR-286 standardına göre geliştirilen bir portlet, bu standardı sağlayan herhangi bir portal sunucusunda çalışabilmektedir. Bu özellik, tez kapsamında irdelenecek olan hızlı portal geliştirimini destekleyen bir özelliktir. Farklı portal sunucularında aynı uygulamaların tekrar tekrar geliştirilmesine gerek kalmadan, portlet uygulaması bir kez geliştirilip farklı uygulamalarda bir çok kez kullanılabilir. , Portal uygulamalarının standartlara bağlı geliştirilmesi uygulamaların yeniden kullanılabilirliğini sağlamaktadır.

- Farklı Servislerin Bir Portal Sayfasında Birlikte Çalışabilmesi: Servlet mimarisinde, bir web sayfasında birden fazla servis aynı anda kullanılamamaktadır. Bu özellik, yeniden kullanılabilirlik seviyesi yüksek uygulamaların gerçekleştirilmesinin önünde büyük bir engel oluşturmaktadır. Portlet Mimarisinde, her portlet veri kaynağı olarak web servisini kullanabilmektedir. Bir portal sayfasında, birden fazla portletin

kullanılabilmesinden dolayı, bir web sayfasında birden fazla web servisinin kullanılabilirliğindedir.

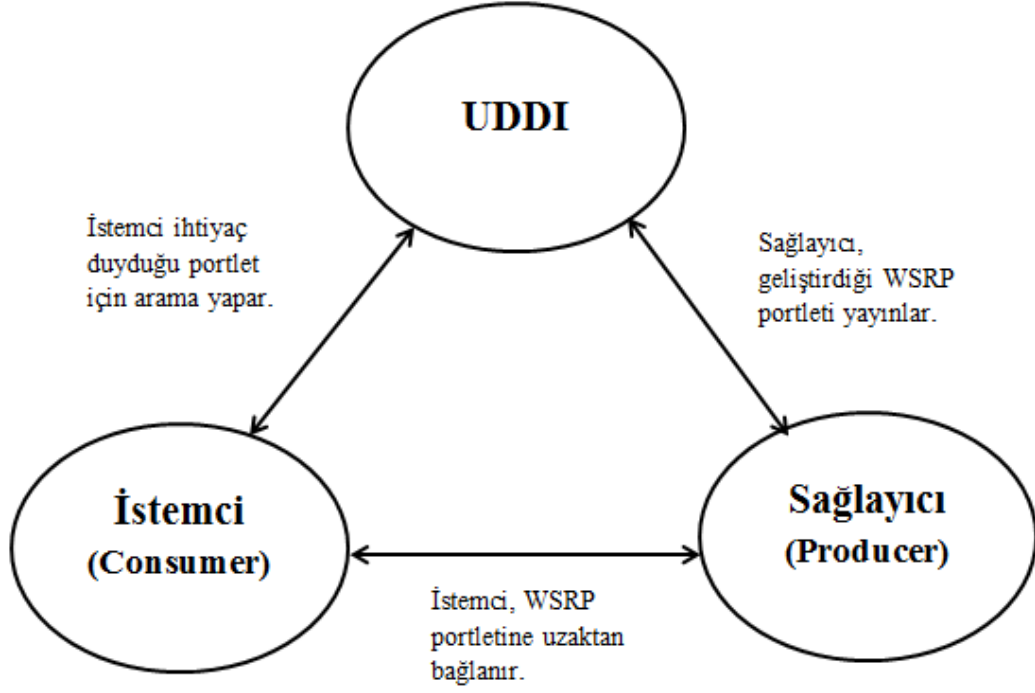
- Kişiselleştirme: Portal geliştiriciler, özel kullanıcı gruplarına özel kişiselleştirilmiş portal sayfaları sunabilmektedir. Aynı portal ortamında farklı kullanıcı gruplarına farklı içerikler sunulabilmektedir. Ayrıca kullanıcılar, portal sayfasında görmek istedikleri içerikleri kendi isteklerine göre kişiselleştirebilmektedir.

- Portletler Arasındaki İletişim: Portal ortamında farklı portlet uygulamaları arasında iletişim sağlanarak, SYM ortamında hala tatmin edici çözüme kavuşturulamamış servis kompozisyonuna en pragmatik çözüm sunulmaktadır. Ayrıca, portletler arasındaki bu koordinasyon sayesinde yeniden kullanılabilirlik seviyesi yüksek uygulamalar geliştirilebilmektedir. Bu koordinasyon sayesinde portal ortamında kompleks uygulamalar kullanıcıların hizmetine sunulabilmektedir.

1.6. Web Services for Remote Portlets belirtimi (WSRP)

Web Services for Remote Portlets belirtimi, OASIS (Organization for the Advancement of Structured Information Standards)'in WSRP Teknik kurulu tarafından tanımlanmıştır. Bu komite tarafından, 2003 yılı eylül ayında WSRP 1.0 standardı yayınlanmış ve 2008 yılı Nisan ayında WSRP 1.0 standardı güncellenmiş ve bu güncellenmiş standart, WSRP 2.0 olarak adlandırılmıştır. OASIS WSRP v2.0 standardı, farklı portal sunucularındaki uygulamaların ve içeriklerin başka bir portal sunucu içerisinde kullanılmasını, programlama bilgisi ve eforu gerektirmeden gerçekleştirmeyi amaçlamaktadır. Bu standart sayesinde farklı portal sunucularındaki portletlerin uzaktan kullanılabilirliği sağlanmakta, Servis Yönelimli Mimaride olduğu gibi dağıtık sistemler arasındaki “birlikte işlerliği” desteklenmektedir.

WSRP standardında sağlayıcı (producer), istemci (consumer), son kullanıcı (end user) ve portlet olmak üzere dört temel aktör tanımlanmıştır. Sağlayıcı (Producer), portletleri geliştirip WSRP olarak yayınlar. İstemci ise yayınlanan WSRP portletine uzaktan bağlantı ile kullanmaktadır. Son kullanıcı, portal üzerinde WSRP portletini kullanan web kullanıcısıdır.



Şekil 12. WSRP aktörleri

Portal ortamı göz önüne alındığında, WSRP sağlayıcısı genellikle “portlet container” olmaktadır. Portal geliştiricisi, JSR 286 standardına göre geliştirdiği portleti, portal yazılımının sağladığı WSRP desteği ile WSRP portleti olarak yayınlatabilmektedir. Başka bir portal geliştirici WSRP olarak yayınlanan bu portleti, istemci olarak uzaktan erişim ile kullanabilmektedir. Sağlayıcı tarafından geliştirilen portletler, WSRP portleti olarak yayınlanır. WSDL tanımları kullanılarak UDDI ya kaydedilir. İstemci taraf ise ihtiyaç duyduğu portlet için sağlayıcının ve portletlerin metadatasını kullanarak arama yapar ve ihtiyaç duyduğu portlete uzaktan erişim sağlar (OASIS, 2008). WSRP standardıyla .NET portalındaki bir .NET portleti, Java portalında uzaktan erişimle kullanılabilir. Bir portal yazılımının WSRP standardını sağlaması üç farklı şekilde gerçekleşmektedir (Yang X., vd., 2006).

- Sadece WSRP istemcisini destekleyen portallar: Bu tür portal ortamlarında, portal geliştiriciler sadece WSRP portlet sağlayıcılarının sunduğu portletleri kullanabilmektedir. Bu portal ortamlarında, geliştiriciler portlet geliştirip bu portletleri WSRP olarak yayınlamamaktadır. uPortal 2.x yazılımı, buna örnek olarak gösterilebilir.

- Sadece WSRP sağlayıcısını destekleyen portallar: Bu tür portal ortamlarında, portal geliştiriciler sadece geliştirdikleri portletleri WSRP olarak yayınlatabilmektedir. Herhangi bir WSRP portletini uzaktan erişim ile kullanamamaktadır. Pluto ve WSRP4J portal yazılımları buna örnek gösterilebilir.
- Sağlayıcı ve istemciyi destekleyen portallar: Bu tarz portal ortamlarında ise portal hem sağlayıcı hemde istemci rolünü gerçekleştirebilmektedir. Liferay ve Exo portal ortamları buna örnek gösterilebilir.

İstemcinin, sağlayıcının geliştirdiği ve WSRP portleti olarak yayınladığı portletlerle iletişim kurulabilmesi için, web servisi arayüzleri sunmaktadır. Servis tanımı (Service Description), Markup, Kayıt (Registration), Portlet yönetimi (Portlet Management), sağlayıcı tarafından sunulan web servisi arayüzleridir.

Servis tanımı arayüzü; WSRP sağlayıcı tarafından sunulması zorunlu bir arayüzdür. İstemci, sağlayıcının yeteneklerini (capabilities) öğrenmek istediği zaman bu arayüzü kullanmaktadır. Bu mimari OGC web servisi mimarisinde de göze çarpmaktadır.

Örneğin; bir Web Feature Servisine (WFS) gönderilen GetCapabilities isteği ile benzerlik taşımaktadır. Servis tanımı arayüzünde, istemci sağlayıcının hangi portletleri sunduğu ve portletler hakkında metaveri almak için kullanılmaktadır.

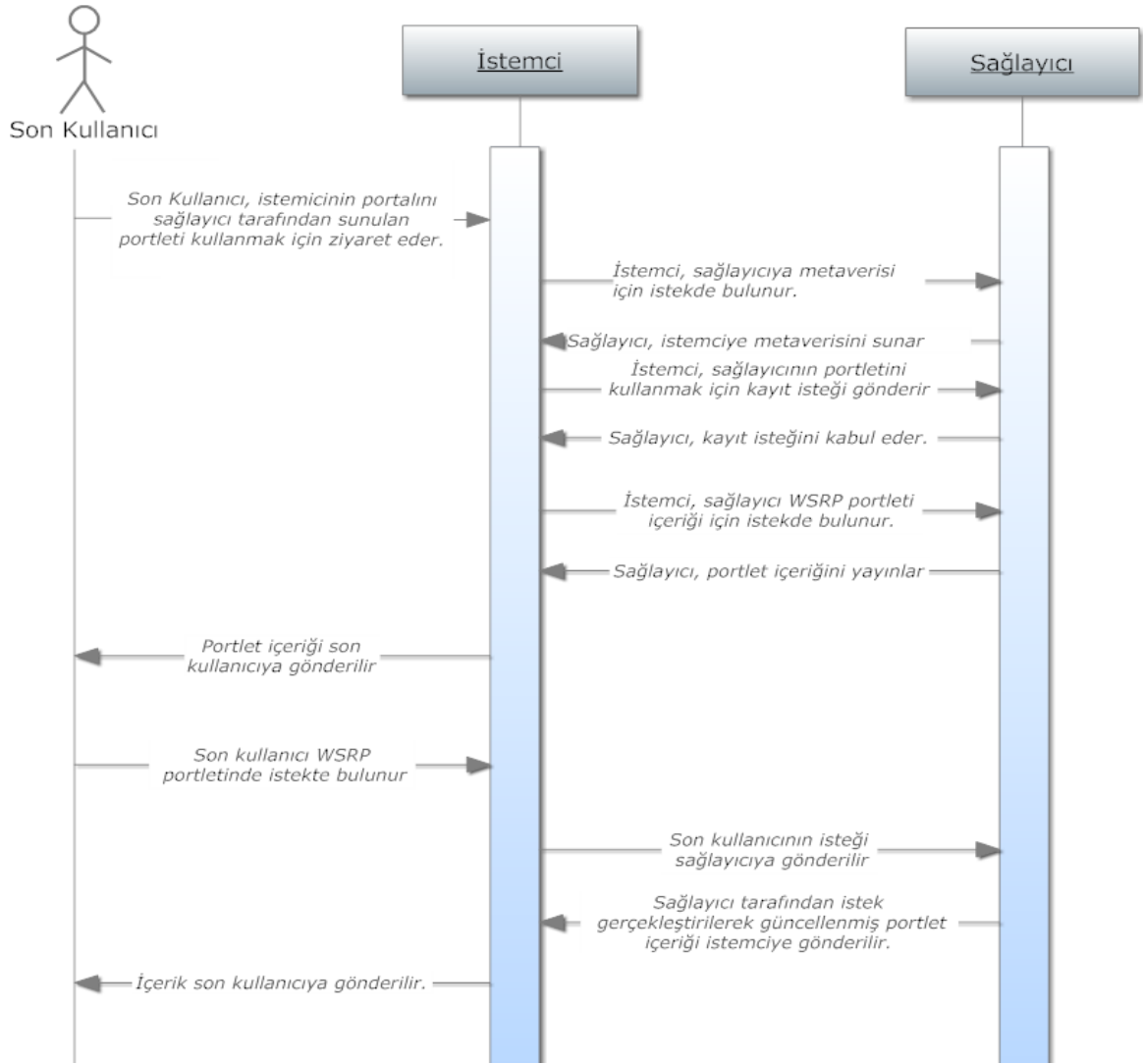
Markup arayüzü; WSRP portletinin içeriğinin yayınlanması bu içerik üzerinde işlem yapılabilmesini sağlayan arayüzdür. Bu arayüz de WSRP sağlayıcısının sağlaması zorunlu bir arayüzdür. Örneğin; WFS sunucusuna gönderilen GetFeature isteği, Markup arayüzü ile benzer özellikler taşımaktadır. Bu arayüz sayesinde istemci uzaktan portlete bağlanarak portlet içeriğini kendi portalında yayınlatabilmektedir.

Kayıt (registration) arayüzü; istemci ve sağlayıcı arasında iletişimin gerçekleştirildiği gerçekleştirilmesi zorunlu olmayan (optional) bir arayüzdür. Bu arayüzün kullanımına örnek olarak, portal üzerinden e-kitap satış işlemi gösterilebilir. Kullanıcının, WSRP portleti üzerinde e-katıp satın alabilmesi için ilk önce kayıt (registration) işlemini gerçekleştirmesi gerekmektedir. Bu gibi durumlarda, bu arayüz kullanılmaktadır.

Portlet Yönetimi arayüzü; sağlayıcının bu arayüzü sağlamasıyla, istemci bu portleti kendi portal sunucusuna kopyalayabilir veya portlet üzerinde değişiklikler yapabilmektedir. OGC servislerinden WFS sunucusuna gönderilen Transaction isteği bu arayüz ile benzerlikler taşımaktadır.

WSRP aktörleri arasındaki iletişimin iş akışı aşağıdaki gibi özetlenebilir.

- İstemci, sağlayıcıyı arama sonucunda bulur. İstemci sağlayıcı hakkındaki metaveriyi inceler, hangi WSRP portletlerini sunduğunu ve kayıt için gerekli bilgileri bu metaveriden öğrenir.
- Kayıt yapılmadan önce istemciye sunulan metaverideki bilgilere göre İstemci ve Sağlayıcı arasında bağlantı kurulur.
- İstemci, kurulan bu bağlantıyla sağlayıcının tüm yetenekleri (capabilities) hakkında bilgi içeren metaveriyi inceler.
- Sağlayıcı, WSRP içeriğini istemciye gönderir.
- Son kullanıcı, WSRP portletini portal ortamında kullanmaya başlar.
- Son kullanıcı, portlet üzerinde herhangi bir işlem yaptığında (örn: form doldurduğunda), istek sağlayıcı tarafından gerçekleştirilir ve tekrar portlet içeriği istemciye gönderir. Son kullanıcıya portlet üzerinden gerçekleştirilen istek sonucunda üretilen içeriği WSRP portletinde gösterilir.



Şekil 13. WSRP iş akışı (Yang X., 2007)

Geliştirilen WSRP portletlerinin, başka kullanıcılar tarafından arama sonucunda bulunabilmesi için UDDI kataloğuna kaydedilmektedir. Kullanıcılar, WSRP portletlerinin WSDL dökümanlarının, UDDI kataloğuna kaydedilmesi ve kullanıcılarının UDDI kataloğundan arama yapmasıyla ihtiyaç duyduğu portletleri bulabilmektedir. Bu WSDL dokümanı, T-model ID, Service Name, Supported locales, Service Titles, Service Descriptions, Supported Modes, Supported Markups, Cachability, Instance awareness, Keywords, Supported View States, Allowed modifications gibi öznitelik verilerini içermektedir (Diaz vd., 2002). UDDI kataloğunda arama yaparken portlet sağlayıcısının değil portlet için arama yapılabilmesi için portlet metadatasının yayınlanması gerekmektedir. Kullanıcı bu şekilde, UDDI kataloğundan arama yapacak ihtiyaç duyduğu portletleri yayınlanan metadatası sayesinde bulabilecektir (Oracle, 2008). Bu yaklaşım portlet

aramalarını standartlaştırılmasını sağlamaktadır. Dolayısıyla, portlet aramaları daha hızlı ve kolay olacaktır.

UKVA ortamında ulusal ve kurum düzeyde portalların geliştirildiği bir ortam düşünüldüğünde, aynı veya benzer uygulamaların portallar için geliştirilmesi söz konusudur. WSRP yaklaşımı bu açıdan önemli bir bileşen olacaktır. Kurumların aynı uygulamaları tekrar geliştirmeleri yerine bu standart sayesinde uzaktan erişim ile bir kurumun portal uygulaması başka bir kurum tarafından kendi portalında kullanılabilir. Bu bağlamda şöyle bir örnek senaryo tasarlanabilir. Bu örnek senaryoda; Ülkemizin ulusal konumsal portalı, JSR-286 ve WSRP standartları kullanılarak geliştirilmiştir. Ulusal Konumsal portaldaki uygulamaların WSRP portleti olarak yayınlanmıştır (Örn: Arama ve Harita Görüntüleyici portletleri). Çevre ve Şehircilik Bakanlığının kurumsal düzeydeki portalında, konumsal veriler için arama hizmeti kullanıcılara sunulmak istenmektedir. Ulusal konumsal portaldan WSRP portleti olarak yayınlanan Arama portleti, uzaktan erişimle bu kurumun portalında kullanılabilir. Kullanıcı, ulusal konumsal portalda arama yapıyormuş gibi Çevre ve Şehircilik Bakanlığının portalından arama yapabilecektir. UKVA ortamında bu bağlamda senaryolar düşünüldüğünde WSRPnin bu konudaki katkısı ortaya çıkmaktadır. Aynı ve benzer uygulamaların tekrar geliştirilmesinin önlenmesiyle, acil ihtiyaç durumunda olan UKVA ortamının oluşturulmasında, ülkemizin kaynak ve zaman kaybının engellenmesine katkıda bulunacaktır.

WSRP mimarisinin öneminin artmasıyla bir çok açık kaynak kodlu portal sunucusu, WSRP çalışmalarını hızlandırmıştır. Açık kaynak kodlu portal sunucularının çoğu WSRP desteklemektedir². Literatürde açık kaynak kodlu yazılımların WSRP destekleri açısından irdelendiği çalışmalar mevcuttur (Yang X. vd., 2007).

1.7 Web Processing Servisi

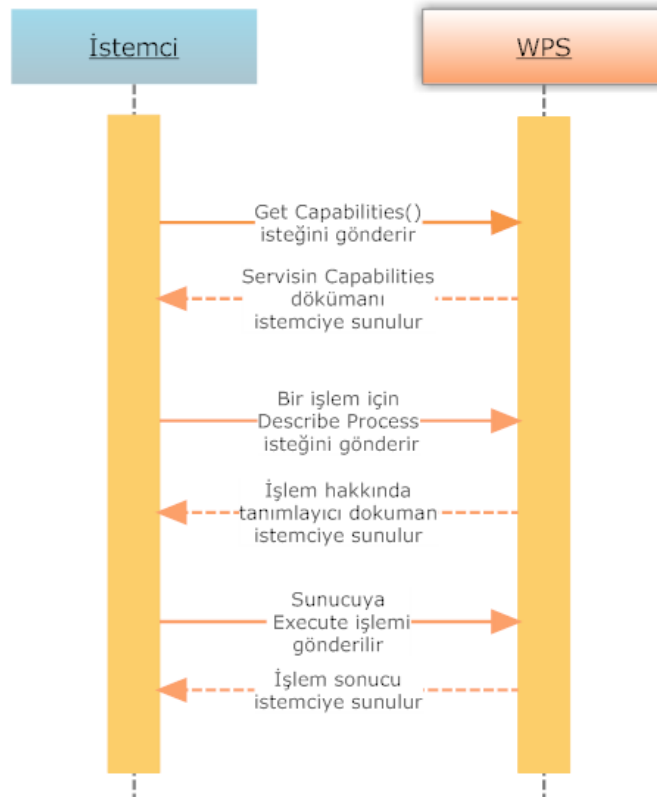
Open Geospatial Consortium (OGC), Web tabanlı konumsal analizlerin, gerçekleştirilmesine yönelik olarak Web Processing Servisi v1.0 (WPS) standardını, 2007 yılında yayınlamıştır. Bu standart ile konumsal WPS işlemlerin girdi çıktı verilerini

² Liferay (URL 9), Exo Platform (URL 10), StringBeans (URL 11), uPortal (URL 12), WSRP4J (URL 13)

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.

Servisin kullanılma prensibi, klasik OGC mimarisindeki gibidir. Kullanıcı ilk önce, Getcapabilities isteği gönderir. Kullanıcı, capabilities dokümanını inceledikten sonra, ihtiyaç duyacağı işlemler için DescribeProcess isteği gönderir ve işlem hakkında ayrıntılı bilgi alır. Daha sonra, sunucuya Execute isteği göndererek işlemi gerçekleştirir (Şekil 14).



Şekil 14. WPS iş akışı

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

WPS isteklerinin sunucuya gönderilmesi, iki yöntem ile yapılmaktadır. Bunlar; KVP encoded ve XML encoded isteklerle, isteklerin gerçekleştirilmesidir. KVP encoded istekler, HTTP GET arayüzü kullanılarak gönderilmektedir. XML encoded istekler ise HTTP POST arayüzü kullanılarak gönderilmektedir.

WPS standardına göre, WPS sunucularının sağlaması gereken encoded istekler Tablo 1’ deki gibidir.

Tablo 1. WPS operasyonları ve sağlaması gereken “encoded” yöntemi

WPS operasyon ismi	Sağlaması gereken encoded yöntemi
GetCapabilities (zorunlu)	KVP (zorunlu), XML (isteğe bağlı)
DescribeProcess (zorunlu)	KVP (zorunlu), XML (isteğe bağlı)
Execute (zorunlu)	XML (zorunlu) ve KVP (isteğe bağlı)

1.7.1 WPS Operasyonları

1.7.1.1 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).

1.7 başlığı adı altında anlatıldığı üzere, bir WPS sunucusu, kullanıcının GetCapabilities operasyonunu KVP encoded isteklerle gerçekleştirmesini desteklemek zorundadır. WPS standardında bir KVP encoded isteklerin nasıl gerçekleştirileceği tanımlanmıştır. Sunucuya gönderilecek KVP encoded istekler, parametrelerden oluşmakta ve her parametre arasında “&” işareti kullanılmaktadır. Tablo 2’ de görüldüğü gibi, standartta GetCapabilities isteği için, iki zorunlu iki de seçmeli olmak üzere dört adet parametre tanımlamıştır. KVP ve XML encoded istekler için aynı parametreler geçerlidir.

Tablo 2. GetCapabilities isteği için tanımlanan parametreler

Parametre adı	Multiplicity	Açıklama
Service	bir (zorunlu)	İstek gönderilecek servisin tanımlandığı parametre
Request	bir (zorunlu)	Kullanıcının servisi kullanarak gerçekleştireceği operasyonunun tanımlandığı parametre
AcceptVersions	sıfır ya da bir (seçmeli)	Servis versiyonunun tanımlandığı parametre
language	sıfır ya da bir (seçmeli)	Servis dilinin tanımlandığı parametre

http://localhost:6161/geoserver/ows?

service=wps&

version=1.0.0&

request=GetCapabilities

Yukarıda örnek bir KVP encoded GetCapabilities isteği görülmektedir. Bu uygulama için Geoserver WPS sunucusu kullanılmıştır.

```
<wps:GetCapabilities xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ows="http://www.opengis.net/ows/1.1" service="WPS"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsGetCapabilities_request.xsd" language="en">
  <wps:AcceptVersions>
    <ows:Version>1.0.0</ows:Version>
    <ows:Version>2.0.0</ows:Version>
    <ows:Version>1.1.0</ows:Version>
  </wps:AcceptVersions>
</wps:GetCapabilities>
```

Yukarıda örnek bir XML encoded GetCapabilities isteği görülmektedir. Bu uygulama için Deegree WPS sunucusu kullanılmıştır.

WPS standardı, GetCapabilities isteđi sonucunda üretilecek Capabilities dokümanının yapısında tanımlamıştır. Tablo 3' te, bir Capabilities dokümanının yapısı gösterilmektedir. Bir WPS kullanıcısı, Capabilities dokümanını bu yapıya göre incelemelidir.

Tablo 3. Capabilities dokümanının yapısı

Parametre Adı	Kardinalite	Açıklama
Service	Bir (zorunlu)	Servis tanımlayıcısı
Version	Bir (zorunlu)	Servisin sağladığı WPS standardı sürümü (Örn: 1.0.0 ya da 0.4.0)
Update Sequence	sıfır ya da bir (seçmeli)	Servis hakkındaki metaveride yapılan değişiklik tarihi, güncelleme tarihi olarak da adlandırılabilir.
lang	Bir (zorunlu)	Servis dil tanımlayıcısı
Service Identification	Bir (zorunlu)	WPS servisi hakkında metaveri
Service Provider	Bir (zorunlu)	Servis sağlayıcısı hakkında bilgi
Operations Metadata	Bir (zorunlu)	Bu WPS servisi tarafından gerçekleştirilen operasyonlar hakkında metaveri
Process Offerings	Bir (zorunlu)	WPS servisi tarafından gerçekleştirilebilen WPS işlemleri hakkında metaveri
Languages	Bir (zorunlu)	WPS servisi tarafından sağlanan diller hakkında metaveri
WSDL	sıfır ya da bir (seçmeli)	WPS servisinin sağladığı operasyonların ve WPS işlemlerinin WSDL tanımlarını içeren WSDL dokümanının konumu

```

<wps:Capabilities service="WPS" version="1.0.0" xml:lang="en" xsi:schemaLocation="
http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsGetCapabilities_response.xsd" xmlns:wps="http://
www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:ogc="h
ttp://www.opengis.net/ogc" xmlns:gml="http://www.opengis.net/gml" xmlns:xlink="http://
www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ows:ServiceIdentification>
    <ows:Title>deegree 3 WPS</ows:Title>
    <ows:Abstract>deegree 3 WPS implementation</ows:Abstract>
    <ows:ServiceType>WPS</ows:ServiceType>
    <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
  </ows:ServiceIdentification>
  ...
  <ows:OperationsMetadata>
    <ows:Operation name="GetCapabilities">
      <ows:DCP>
        <ows:HTTP>
          <ows:Get xlink:href="http://localhost:8080/deegree-wps-
demo/services?"></ows:Get>
          <ows:Post xlink:href="http://localhost:8080/deegree-wps-
demo/services"></ows:Post>
        </ows:HTTP>
      </ows:DCP>
    </ows:Operation>
  ..
  <wps:ProcessOfferings>
    <wps:Process wps:processVersion="1.0.0">
      <ows:Identifier>BufferEmre</ows:Identifier>
      <ows:Title>WFS tarafından sunulan GMLlere buffer uygulayabilen process gml
stream</ows:Title>
      <ows:Abstract>Bütün GML versiyonlarıyla çalışır.</ows:Abstract>
    </wps:Process>

```

Yukarıda bir WPS servisinin Capabilities dokümanının bir kısmını görülmektedir. Bu uygulama için DeegreeWPS sunucusu kullanılmıştır. Ek 3’de örnek GetCapabilities dokümanı mevcuttur.

1.7.1.2 DescribeProcess Operasyonu

DescribeProcess operasyonu, WPS sunucusunun sunduğu işlemin kullanılabilmesi 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 alınacaksa, o işlemin tanımlayıcısı (identifier) kullanılarak gerçekleştirilmektedir.

1.7 başlığı adı altında anlatıldığı üzere, bir WPS sunucusu, kullanıcının DescribeProcess operasyonunu KVP encoded isteklerle gerçekleştirmesini desteklemek zorundadır. Tablo 4’ te görüldüğü gibi, standartta DescribeProcess isteği için, dört zorunlu bir de seçmeli olmak üzere beş adet parametre tanımlanmıştır.

Tablo 4. DescribeProcess isteği için tanımlanmış parametreler

Parametre adı	Multiplicity	Açıklama
service	bir (zorunlu)	İstek gönderilecek servisin tanımlandığı parametre
request	bir (zorunlu)	Kullanıcının servisi kullanarak gerçekleştireceği operasyonunun tanımlandığı parametre
version	bir (zorunlu)	Servis versiyonunun tanımlandığı parametre
language	sıfır ya da bir (seçmeli)	Servis dilinin tanımlandığı parametre
Identifier	Bir ya da daha fazla (zorunlu)	Hakkında bilgi alınacak WPS işlemlerinin tanımlayıcısı

http://localhost:8080/deegree-wps-demo/services?

service=WPS&

VERSION=1.0.0&

request=DescribeProcess&

IDENTIFIER=BufferEmre

Yukarıda örnek bir KVP encoded DescribeProcess isteği görülmektedir. Bu uygulama için Deegree WPS sunucusu kullanılmıştır. Sunucuya, birden fazla WPS işlemi için DescribeProcess isteği aynı anda gönderilebilmektedir.

<DescribeProcess xmlns="http://www.opengis.net/wps/1.0.0"

xmlns:xlink="http://www.w3.org/1999/xlink"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:ows="http://www.opengis.net/ows/1.1"

xsi:schemaLocation="http://www.opengis.net/wps/1.0.0"

```

http://schemas.opengis.net/wps/1.0.0/wpsDescribeProcess_request.xsd" service="WPS"
version="1.0.0" language="en">

  <ows:Identifier>BufferEmre</ows:Identifier>

</DescribeProcess>

```

Yukarıda örnek bir XML encoded GetCapabilities isteği görülmektedir. Bu uygulama için Deegree WPS sunucusu kullanılmıştır.

WPS standardı, DescribeProcess isteği sonucunda üretilen dokümanın yapısında tanımlanmıştır. Tablo 5'te, bir DescribeProcess dokümanın yapısı gösterilmektedir. Bir WPS kullanıcısı, DescribeProcess dokümanını bu yapıya göre incelemelidir.

Tablo 5. DescribeProcess dokümanının yapısı

Parametre Adı	Multiplicity and Use	Açıklama
Identifier	Bir (zorunlu)	WPS işlemi hakkında tanımlayıcı bilgiler
Title	Bir (zorunlu)	
Abstract	sıfır ya da bir (seçmeli)	
Metadata	sıfır ya da bir (seçmeli)	
Profile	sıfır ya da bir (seçmeli)	
processVersion	Bir (zorunlu)	
WSDL	sıfır ya da bir (seçmeli)	
DataInputs	sıfır ya da bir (seçmeli) ³	WPS işleminin gerçekleştirilmesi için gerekli girdi verileri
ProcessOutputs	Bir (zorunlu)	WPS işlemi sonucunda oluşacak veri hakkında tanımlayıcı bilgiler

³ Her WPS işlemi, aslında bir girdi verisine, zorunlu olarak ihtiyaç duymaktadır. Fakat, bazı WPS işlemleri sabit (fixed) input verilerini kullanabilmektedir. Dolayısıyla, kullanıcının girdi verisi tanımlamadığı WPS işlemleri de olabilmektedir.

Tablo 5'in devamı

storeSupported	sıfır ya da bir (seçmeli)	Bir WPS işleminin gerçekleştirilmesi sonucu üretilecek verinin sunucu tarafından web üzerinden ulaşılabilir bir kaynak tarafından sağlanıp sağlanmaması durumudur.
statusSupported	sıfır ya da bir (seçmeli)	

```

<wps:ProcessDescriptions service="WPS" version="1.0.0" xml:lang="en" ...>
  <ProcessDescription wps:processVersion="1.0.0" storeSupported="true" statusSupported="false">
    <ows:Identifier>BufferEmre</ows:Identifier>
    <ows:Title>WFS tarafından sunulan GMLlere buffer uygulayabilen process gml stream</ows:Title>
    <ows:Abstract>Bütün GML versiyonlarıyla çalışır.</ows:Abstract>
    <DataInputs>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier>GMLInput</ows:Identifier>
        <ows:Title>GMLInput</ows:Title>
        <ComplexData>
          <Default>
            <Format>
              <MimeType>text/xml; subtype=gml/3.1.1</MimeType>
            </Format>
          </Default>...
        </ComplexData>
      </Input>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier>BufferDistance</ows:Identifier>
        <ows:Title>Buffer distance</ows:Title>
        <LiteralData>... </LiteralData>
      </Input>
    </DataInputs>
    <ProcessOutputs>
      <Output>
        <ows:Identifier>BufferedGML</ows:Identifier>
        <ows:Title>BufferedGML</ows:Title>...
      </Output>
    </ProcessOutputs>
  </ProcessDescription>
</wps:ProcessDescriptions>

```

Yukarıda bir WPS servisinin Capabilities dokümanının bir kısmını görülmektedir. Bu uygulama için DegreeWPS sunucusu kullanılmıştır. Ek 4’de örnek DescribeProcess dokümanı mevcuttur.

1.7.1.3 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 (identifier) kullanarak, girdi verilerini ve çıktı verilerini tanımlayarak veri üzerinde işlem yapabilmek için *Execute* operasyonunu kullanır. Ek 5’te örnek Execute dokümanı mevcuttur.

1.7 başlığı adı altında anlatıldığı üzere, bir WPS sunucusu, kullanıcının Execute operasyonunu XML encoded isteklerle gerçekleştirmesini desteklemek zorundadır. Tablo 6’da görüldüğü gibi, standartta Execute isteği için, dört zorunlu üç seçmeli olmak üzere beş adet parametre tanımlanmıştır.

Tablo 6. Execute isteği için tanımlanmış parametreler

Parametre adı	Multiplicity	Açıklama
Service	Bir (zorunlu)	İstek gönderilecek servisin tanımlandığı parametre
Request	Bir (zorunlu)	Kullanıcının, servisi kullanarak gerçekleştireceği operasyonunun tanımlandığı parametre
Version	Bir (zorunlu)	Servis versiyonunun tanımlandığı parametre
Identifier	Bir (zorunlu)	Kullanılacak WPS işleminin tanımlayıcısı
DataInputs	sıfır ya da bir (seçmeli)	WPS işleminin gerçekleştirilmesi için gerekli girdi verilerinin tanımlandığı parametre

Tablo 6' nin devamı

Response Form	sıfır ya da bir (seçmeli)	WPS işlemi sonucunda üretilecek çıktı dokümanının formatının tanımlandığı parametre
Language	Bir (seçmeli)	WPS servis dilinin tanımlandığı parametre

```

<wps:Execute ... service="WPS" version="1.0.0" ...>
  <ows:Identifier>Buffer</ows:Identifier>
  <wps>DataInputs>
    <wps:Input>
      <ows:Identifier>GMLInput</ows:Identifier>
      <wps>Data>
        ...
      </wps>Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>BufferDistance</ows:Identifier>
      ...
    </wps:Input>
  </wps>DataInputs>
  <wps:ResponseForm>
    ...
  </wps:ResponseForm>
</wps:Execute>

```

Yukarıda örnek bir XML encoded Execute isteği görülmektedir. Bu uygulama için Deegree WPS sunucusu kullanılmıştır. Ek 5' te bu Execute isteğinin tamamı mevcuttur. KVP encoded isteklerle Execute isteğinin gerçekleştirilmesi, WPS standardıyla zorunlu koşulmadığı için bir çok WPS sunucusu bu desteği sağlamamıştır. Geoserver ve 52 North WPS bu desteği sağlayan, WPS sunucularındandır.

```

http://localhost:6161/geoserver/ows?service=wps&
version=1.0.0&
request=Execute&
Identifier=JTS: Buffer&

```


DataInputs=features=@mimeType=text/xml@href=http://localhost:/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=bufferedGML@mimeType=text/xml

Yukarıda örnek bir KVP encoded Execute isteği görülmektedir. Bu uygulama için Geoserver WPS sunucusu kullanılmıştır. Bu Execute isteğinde, kullanıcı bir WFS sunucusu tarafından sağlanan yol verisine buffer analizi uygulamaktadır.

WPS standardı, Execute isteği sonucunda üretilen dokümanın yapısında tanımlanmıştır. Tablo 7’ de, bir Execute dokümanın yapısı gösterilmektedir. Bir WPS kullanıcısı, Execute dokümanını bu yapıya göre incelemelidir.

Tablo 7. Execute isteği sonucunda üretilen dokümanın yapısı

Parametre Adı	Multiplicity and Use	Açıklama
Service	Bir (zorunlu)	İstek gönderilecek servisin tanımlandığı parametre
Version	Bir (zorunlu)	Servis versiyonunun tanımlandığı parametre
Lang	Bir (zorunlu)	WPS servis dilinin tanımlandığı parametre
Status Location	sıfır ya da bir (seçmeli)	Execute operasyonunun sonucunda WPS sunucusunun ürettiği dokümanın bulunduğu konumun tanımlandığı parametre
Service Instance	Bir (zorunlu)	Söz konusu WPS servisinin GetCapabilities “url”si
Process	Bir (zorunlu)	Process hakkında tanımlayıcı bilgi içeren parametre
Status	Bir (zorunlu)	Söz konusu işlemin durumunu (tamamlanıp-tamamlanmadığı) belirten parametre
DataInputs	sıfır ya da bir (seçmeli)	Söz konusu WPS işlemi için kullanılan girdi verilerin tanımını içeren parametre

Tablo 7' nin devamı

Output Definitions	sıfır ya da bir (seçmeli)	WPS işlemi sonucunda üretilen çıktı veri hakkındaki tanımlayıcı bilgileri içeren parametre
Process Outputs	sıfır ya da bir (seçmeli)	WPS işlemi sonucunda üretilen veriyi içeren parametre

```

<wps:ExecuteResponse service="WPS" version="1.0.0" xml:lang="en" ... serviceInstance
="http://localhost:8080/deegree-wps-demo/services?service=WPS&request=
GetCapabilities&version=1.0.0" ...>
  <wps:Process wps:processVersion="1.0.0">
    <ows:Identifier>Buffer</ows:Identifier>
  ...
  </wps:Process>
  <wps:Status creationTime="2011-12-22T15:32:16.066Z">
  ...
  </wps:Status>
  <wps:ProcessOutputs>
    <wps:Output>
      <ows:Identifier>BufferedGML</ows:Identifier>
      <ows:Title>BufferedGML</ows:Title>
    ...
    </wps:Output>
  </wps:ProcessOutputs>
</wps:ExecuteResponse>

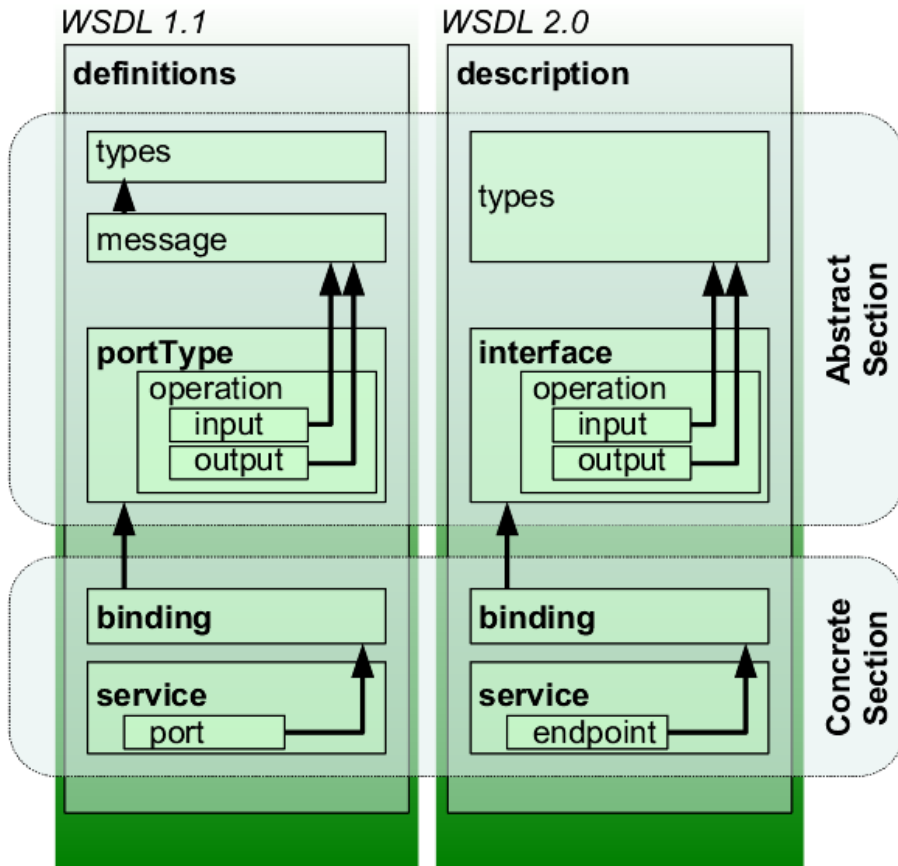
```

Yukarıda, Execute operasyonu sonucunda üretilen örnek sonuç dokümanın bir kısmı görülmektedir. Ek 6' da bu sonuç dokümanının tamamı, mevcuttur.

Bu çalışmada WPS, geliştirilecek konumsal portal ortamında, konumsal veri üzerinde analizlerin, web üzerinden gerçekleştirilmesi aşamasında kullanılacaktır. Sonuç olarak, OGC tarafından tanımlanan WPS standardı, kullanıcıların web üzerinden konumsal veri üzerinde, işlem yapabilmesine yönelik hazırlanmıştır. Klasik OGC mimarisinde olduğu gibi bir işleyiş söz konusudur. Bu çalışma kapsamında, WPS standardı ve WPS sunucuları incelenmiştir. Bu inceleme sonucunda, çeşitli eksiklikler tespit edilmiştir. İlerleyen bölümlerde, ayrıntılı bir şekilde bu sorunlar dile getirilecek ve çözüm önerileri ortaya koyulacaktır.

1.8. Web Servisleri Tanımlama Dili

Web Servisleri Tanımlama Dili (Web Services Description Language / WSDL), Web servislerini tanımlamak için geliştirilen XML tabanlı bir dildir. WSDL belirtiminin ilk sürümü (version 1.0), Microsoft, IBM ve Ariba yazılım firmaları tarafından Eylül 2000 de geliştirilmiştir. Adı geçen yazılım firmaları, Mart 2001 de belirtimin ikinci sürümünü(version 1.1) geliştirmiş ve standart olarak kabul edilmesi için W3C ortaklığına sunmuştur (Akıncı, 2006). W3C, WSDL belirtimini aynı tarihte “W3C notu” (W3C Note) olarak yayınlamıştır (W3C, 2001). WSDL belirtiminin son sürümü (version 2.0), W3C tarafından Mart 2006 da yayınlanmıştır (W3C, 2006). WSDL belirtimi W3C tarafından henüz standart olarak kabul edilmemiştir. W3C nin WSDL çalışma grubu belirtim üzerindeki çalışmalarına devam etmektedir (Akıncı, 2006). WSDL, Web servislerinin arayüzlerini tanımlayan, platformdan ve dilden bağımsız bir dildir. WSDL, herhangi bir programlama dili kullanılarak gerçekleştirilen ve herhangi bir platform üzerinden sunulan Web servislerini tanımlama yeteneğine sahiptir (Nagappan vd., 2003).



Şekil 15. WSDL sürümlerine göre doküman yapısı(URL 14)

Bir WSDL dokümanı, *definitions*, *types*, *message*, *portType*, *binding* ve *service* olarak adlandırılan altı element içermektedir. Ek 6 da örnek WSDL dokümanı mevcuttur.

Bu çalışmada WSDL, çalışma kapsamında geliştirilen WPS işlemlerinin WSDL tanımlarını içeren WSDL dokümanlarının hazırlanmasında kullanılmıştır. WPS işlemlerinin WSDL tanımları kullanılarak BPEL ve İş akışı motorları ile servis zincirleme uygulamaları yapılmıştır. İlerleyen bölümler bu uygulama hakkında ayrıntılı bilgiye yer verilecektir.

2. YAPILAN ÇALIŞMALAR

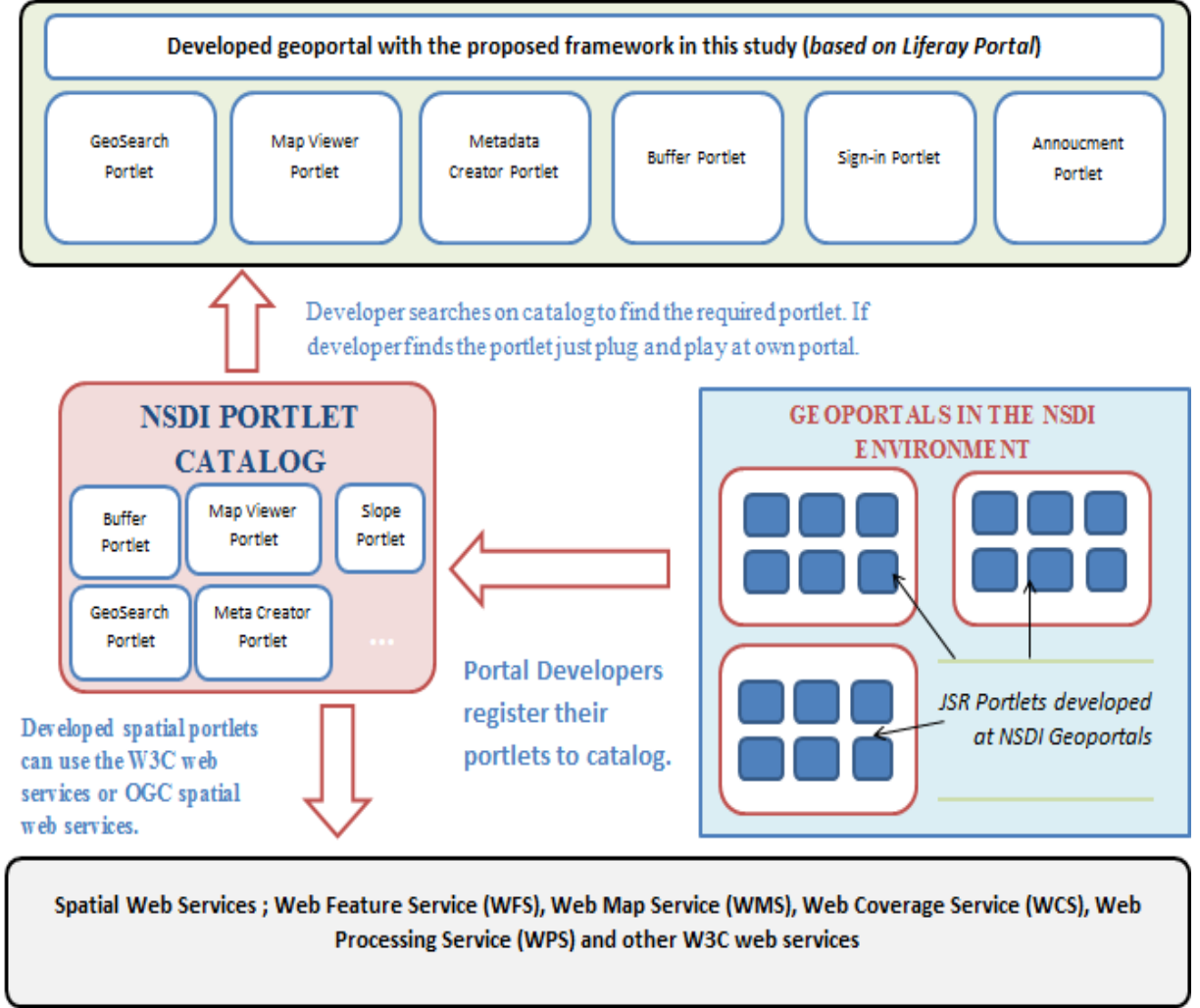
2.1 UKVA Konumsal Portal Çerçevesi

Bu çalışmada, UKVA ortamında konumsal portal geliştirilmesine yönelik bir çerçeve ortaya koyulmuştur (Şekil 16). Bu çerçeve, konumsal portal geliştirimine yönelik çözüm önerileri içermektedir. Konumsal portallara yönelik, yeni bileşenler içermektedir. UKVA ortamında konumsal portalların nasıl geliştirilmesi gerektiği ortaya koyulmuş ve bu doğrultuda prototip konumsal portallar geliştirilmiştir. Bu çerçeveyi, UKVA ortamında hızlı portal geliştirimi, portal ortamında web servisleri kompozisyonunun sağlanmasıyla uygulama tabanlı konumsal portal geliştirimi, UKVA ortamında açık ve özgür kaynak kodlu portal uygulamaları, olarak ana üç bileşenden oluşmaktadır.

Konumsal portallar için yeni yaklaşımlardan bir tanesi olan UKVA ortamında hızlı portal geliştirimi, Türkiye gibi UKVA çalışmalarının gerisinde kalmış ülkeler için, büyük önem arz etmektedir. Hiç şüphe yok ki, UKVA ortamında çok sayıda konumsal portal geliştirilecektir. Aynı veya benzer uygulamalar her portal için tekrar tekrar geliştirilecektir. Yeniden kullanılabilir olmayan uygulamalar, zaman ve kaynak kaybına yol açacaktır. Bu çalışmada önerilen çerçeve, portal uygulamalarının yeniden kullanılabilirlik sorununa çözüm olacaktır.

Bu çerçevenin hızlı portal geliştirimi için ortaya koyduğu bileşenlerden bir tanesi, konumsal portal uygulamalarının geliştirilmesinde standartların kullanılmasıdır. Bu bileşen, konumsal portal uygulamalarının geliştirilmesinde, JSR-286 ve WSRP standartlarının kullanılmasıyla gerçekleştirilecektir. Portlet geliştirimi belli standartlara göre yapılması oluşturulan portletlerin yeniden kullanılabilmesini ve standartları sağlayan herhangi bir portal sunucusunda çalışmasını sağlamaktadır. JSR 286 standardı, portletler için “Portlet API v2.0” tanımlayarak, portletler ve portallar arasındaki “yeniden kullanılabilirliği” sağlamayı amaçlamaktadır. JSR 286 standardını sağlayan bütün portletler bu standartları sağlayan tüm portal sunucularında çalışabilmektedir. WSRP standardı, farklı portal sunucuları arasındaki birlikte işlerliği sağlamayı amaçlamaktadır. Bu standart sayesinde, bir portal sunusundan WSRP portleti olarak yayınlan bir portlet, başka bir portal sunucusu tarafından, uzaktan erişim ile kullanılabilir.

Diğer bir bileşen ise, “UKVA Portlet Kataloğu” bileşenidir. Konumsal portal geliştiricileri, ihtiyaç duydukları portlet uygulamaları için bu katalogdan arama yapacaktır. Eğer ihtiyaç duyulan uygulamaya benzer veya aynı uygulama bulunursa, bu katalog üzerinden portlet uygulaması indirilebilecek ve standartları sağlayan herhangi bir portal sunucusunda kullanılabilir. Bu portlet kataloğunda, açık kaynak kodlu veya ticari portlet uygulamaları olabilecektir.

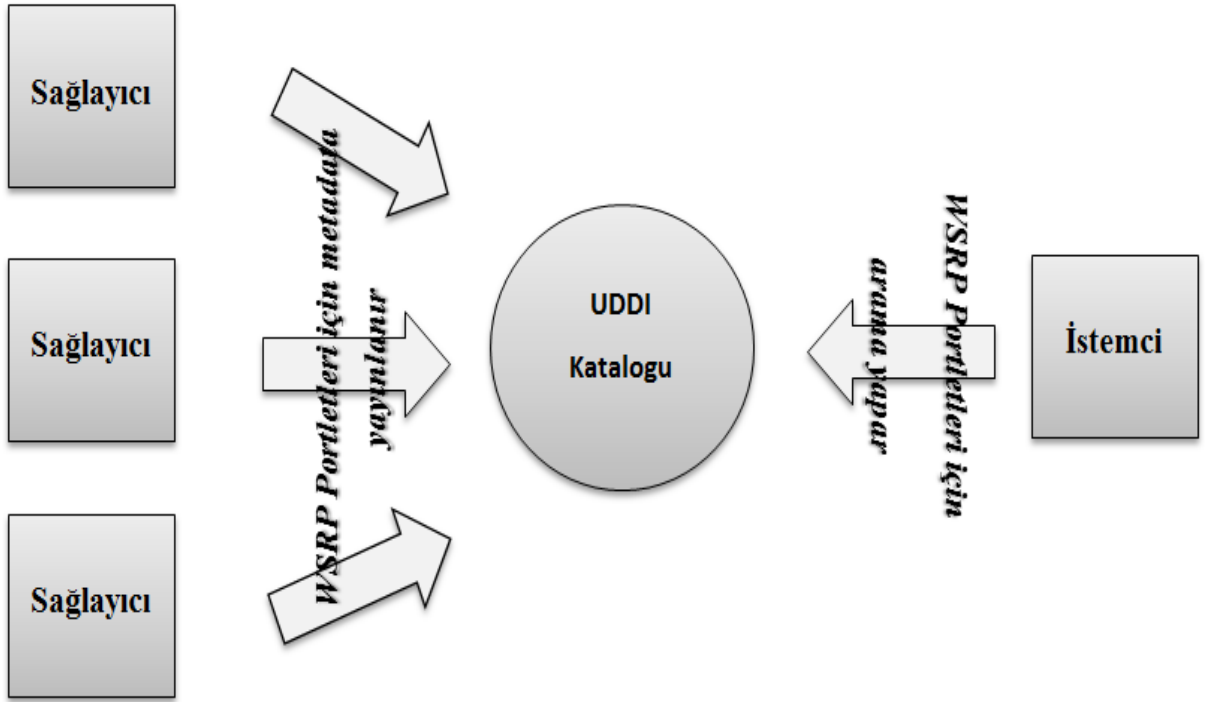


Şekil 16. UKVA Konumsal portal geliştirimi için ortaya koyulan çerçeve

Uygulama geliştiricileri, geliştirdikleri uygulamaları, bu katalog üzerinde açık kaynak kodlu olarak yayınlatabilecektir. Bu tarzdaki portlet uygulamaları, farklı konumsal portallar için geliştirilecek portal uygulamaları için başlangıç noktası olabilecektir. Geliştirici, açık kaynak kodlu portlet uygulamalarını indirip, üzerinde geliştirmeler yaparak farklı portal uygulamaları için uygulama geliştirebilecektir. Geliştirdiği bu portlet uygulamasını yine bu portlet kataloğundan yayınlatabilecektir. Ticari portlet uygulamaları

ise konumsal portallar için daha karmaşık ve profesyonel çözümler sunabilecektir. Bu yaklaşımla özel sektör, UKVA ortamında konumsal portallar için daha karmaşık ve profesyonel çözümler sunabilecektir. UKVA portlet kataloğunda yayınlanan, açık ve özgür kaynak kodlu uygulamalar, özel sektör yazılım grupları tarafından geliştirilerek UKVA ortamındaki sorunlara yönelik nokta çözümlerle uygulamalar geliştirebileceklerdir.

UKVA ortamında, kurumlar arasında portlet paylaşımının gerçekleştirilmesi, portletlerin WSRP olarak yayınlanması ve WSDL tanımları ile UDDI kataloğuna kaydedilerek, kullanıcıların bu katalog üzerinden arama yaparak WSRP portletine uzaktan erişimle de sağlanabilecektir (Şekil 17). Fakat portletlerin WSRP ile yayınlanması, portlet kataloğuna göre dezavantaja sahiptir. Bu yaklaşımda, kullanıcı, sağlayıcının WSRP olarak yayınlandığı portleti aynen kullanmak zorundadır. WSRP yaklaşımı ile portlet üzerinde herhangi bir değişiklik yapılamamaktadır. Bu çerçeveden getirilen portlet kataloğu, WSRP yaklaşımının bu dezavantajını da ortadan kaldırmaktadır. Geliştirilen WSRP portletinin, öngörülen portlet kataloğuna kaydedilmesi ile WSRP portletinin geliştirilmesi sağlanacaktır. WSRP yaklaşımı ile portlet kataloğu yaklaşımı birlikte kullanılarak, bu dezavantaj ortadan kaldırılabilir.



Şekil 17. UDDI kataloğuna portletlerin kaydedilmesi (Oracle, 2008)

UKVA ortamında kurumlar bazında geliştirilen portlet tabanlı web uygulamalarının paylaşımı için portlet katalogları oluşturulduğu ortamda ilgili kurumların portal üretim aşamaları şu şekilde olacaktır. İlgili birimdeki kullanıcı portlet kataloglarından arama yapar. İhtiyacını karşılayacak portletleri bulduğunda katalogdan portlet dosyasını indirir. Portalın portlet yükleme arayüzünden, portletleri portala yükler. Portleti, portal sayfasında uygun bir yere yerleştirerek, portal sayfasını kaydeder ve portal sayfasını oluşturur. Çevre ve Orman Bakanlığındaki ilgili birim, portal geliştirimini gerçekleştirmiş ve portlet kataloğunda harita görüntüleyici portletini yayınlamışsa, Bayındırlık ve İskân Bakanlığında ilgili birimdeki kullanıcı, bu katalogdan harita görüntüleyici portletini indirebilecek ve portalına ekleyebilecektir. Bu şekilde UKVA ortamında hızlı portal geliştirimi sağlanacaktır.

Açık kaynak kodlu ve ticari portal yazılımlarının, hızlı portal geliştirimi için ortaya koyulan çerçevede dile getirilen yaklaşımlardan biri olan portlet kataloğuna benzer portlet katalogları mevcuttur¹. Portal geliştiriciler bu kataloglardan ihtiyaç duydukları portletlerin dosyalarını indirerek portlet yükleme arayüzünden portal sunucusuna yükleyebilmektedir. Fakat bu portlet katalogları incelendiğinde bu kataloglarda herhangi bir standarda göre arama yapılamıyor olması, UKVA ortamı düşünüldüğünde, çok sayıda portlet uygulaması arasından ihtiyaç duyulan portletin bulunmasını zorlaştıracak, hatta imkânsız hale getirecektir. Dolayısıyla portletlerin kataloğa kaydedilmesi ve yayınlanması için portlet hakkında metaveri dokümanlarının oluşturulmasında standardın sağlanması gerekmektedir. Bu eksiklik, portlet dosyasının içindeki, portlet hakkındaki bilgileri içeren Portlet.xml dosyasının bu katalog üzerinden yayınlanması ile giderilebilecektir. Geliştirilen örnek portletin Portlet.xml dosyası aşağıda görülmektedir. Portlet.xml dosyasının tamamı Ek 2’ de mevcuttur.

¹ Liferay Portlet Kataloğu, <http://www.liferay.com/community/wiki/-/wiki/Main/Liferay+Portlets>
 SyncEX Portlet Kataloğu, http://www.syncex.com/portlet_catalog/portlet_catalog.htm
 IBM Lotus and WebSphere Portal Business Solutions Catalog, <https://greenhouse.lotus.com/>
 Jetspeed Portlet Kataloğu, <http://portals.apache.org/jetspeed-1/catalog.html>


```

...
<portlet>
  <description>Geocoder portleti, verilen adresin coğrafi koordinatlara
  dönüştürmektedir.</description>

  <portlet-name>Geocoder</portlet-name>

  <display-name>Geocoder</display-name>

  <portlet-class>com.test.Geocoder</portlet-class>

  <expiration-cache>0</expiration-cache>
  <supports>
    <mime-type>text/html</mime-type>
    <portlet-mode>VIEW</portlet-mode>
    <portlet-mode>EDIT</portlet-mode>
    <portlet-mode>HELP</portlet-mode>
  </supports>
  <resource-bundle>com.test.messages</resource-bundle>

  <portlet-info>
    <title>Geocoder</title>
    <short-title>Geocoder</short-title>
  </portlet-info>

  <supported-publishing-event xmlns:x='http://sunevents.com'>
    <qname>x:geocoderevent</qname>
  </supported-publishing-event>
</portlet>
...

```

Portlet.xml dosyasının metaveri olarak portlet kataloğuna kaydedilmesiyle, portlet aramaları için belli bir standardın sağlanmasıyla, ihtiyaç duyulan portletin arama sonuçlarında hızlı ve kolay bir şekilde bulunabilmesi sağlanacaktır.

Bu çalışmada ortaya koyulan çerçevenin getirdiği yeni yaklaşımlardan bir diğeri ise, portal ortamında web servisleri kompozisyonunun sağlanmasıyla uygulama tabanlı konumsal portal geliştirmedir. Web servisleri kompozisyonu, SYM'nin asıl katkı noktalarından bir tanesi olması bakımından önemli bir konudur. Portal ve portlet teknolojisi ise etkin bir SYM ortamına geçiş sürecinde ara geçiş evresi olarak gözükmemektedir (Akram A, vd., 2005). WSK, bir işlemin gerçekleştirilmesi için birden fazla servisinin bir arada kullanılması olarak tanımlanmaktadır. Bu çalışmanın amaçlarından bir

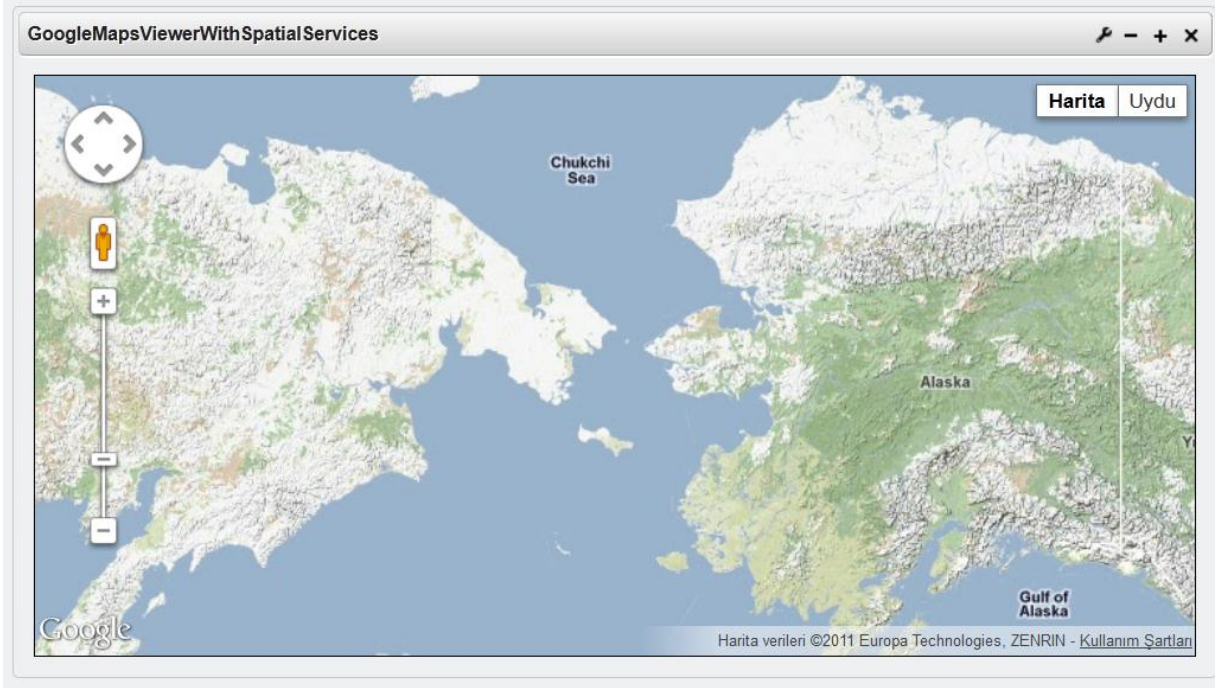
tanesi de, Portal ortamının da WSK' nin gerçekleştirilmesi ile uygulama tabanlı portal ortamının yaratılmasıdır.

Konumsal portalların büyük çoğunluğunun katalog konumsal portalı olması ve bu yapıdaki konumsal portallar, UKVA ile hedeflenen birlikte işlerlik mekanizmalarının etkin bir şekilde çalışmasına tam anlamıyla olanak sağlayamamaktadır. Kullanıcıların, bu yapıdaki konumsal portaldan, uygulama gerçekleştirmek için gereksinim duydukları veri için arama yaptığı süre bazen saatleri almakta, bazende istenilen veriye ulaşmayı mümkün kılmamaktadır. Dolayısıyla, ihtiyaç duyulan verinin portal üzerinden sunulması yerine, kullanıcının portal üzerinden ihtiyaç duyduğu uygulamayı gerçekleştirebildiği, bir konumsal portal ortamı yaratılması gerekmektedir.

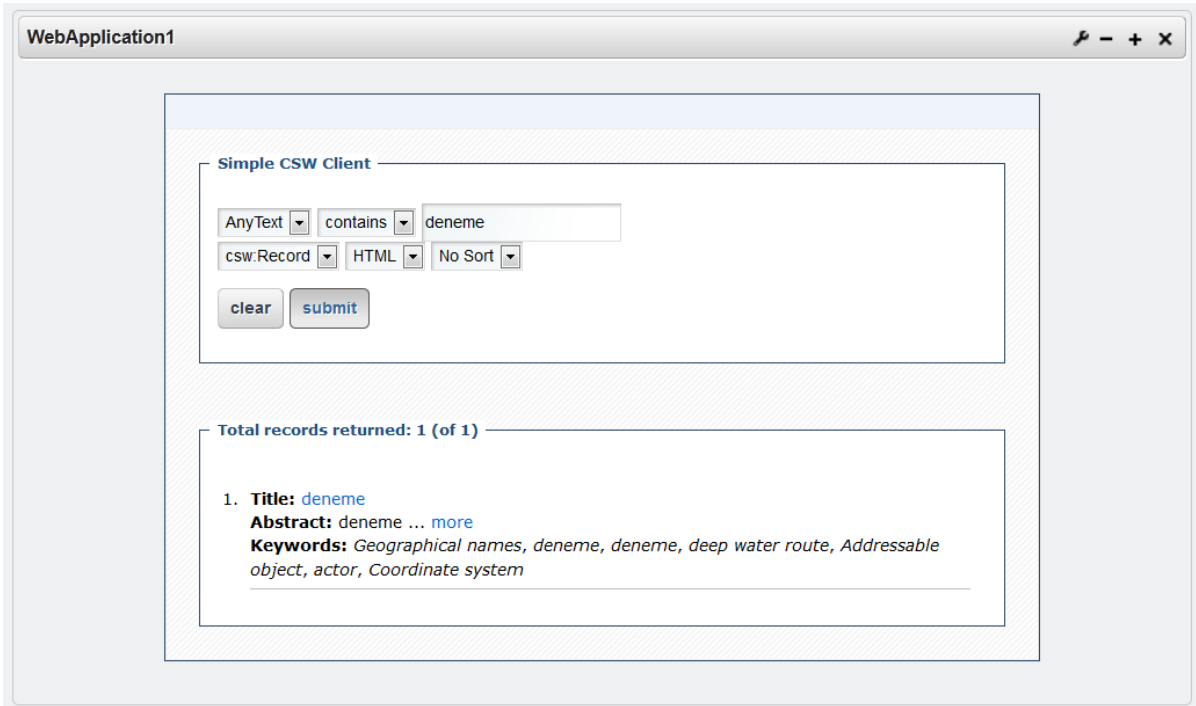
Bu yaklaşım da, konumsal analizlerin portal ortamında gerçekleştirilebildiği bir portal ortamı içermektedir. Web üzerinden konumsal analizler, OGC tarafından 2007 yılında yayınlanan Web Processing Servisi v1.0 kullanılarak gerçekleştirilecektir.

2.2. Yeniden Kullanılabilir Portlet Tabanlı Prototip Konumsal Portal Geliştirimi

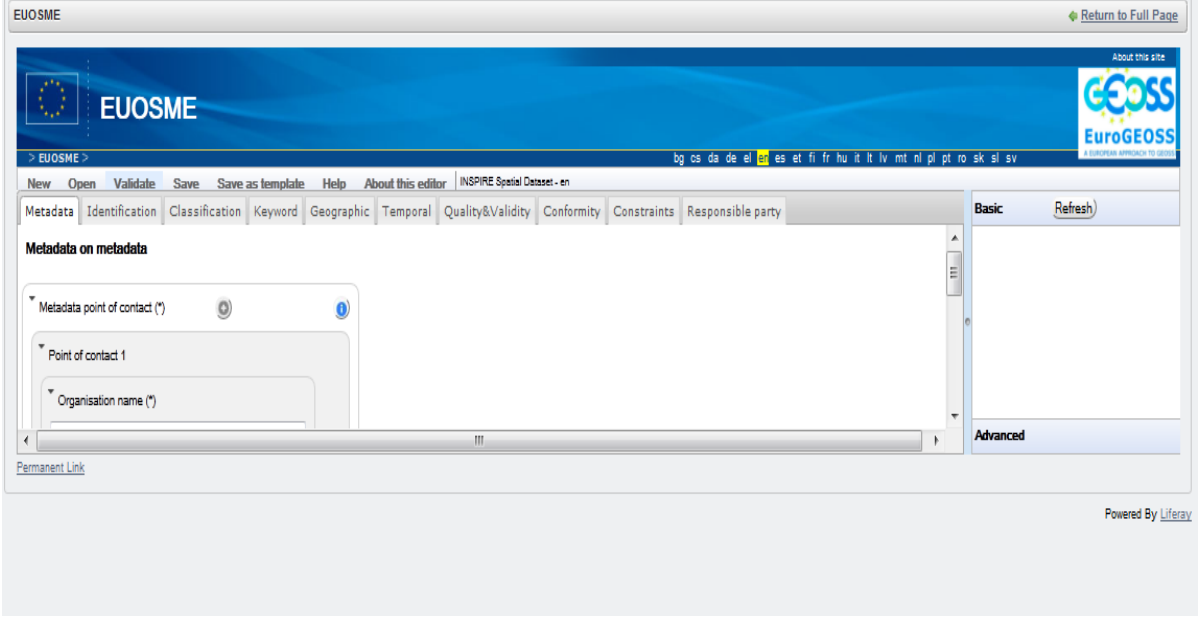
Bu çalışmayla, konumsal portalların geliştirimi için bir yaklaşım ortaya konulmaktadır. Bu yaklaşım kapsamında, “yeniden kullanılabilir” portletlerle, prototip konumsal portal geliştirilmiştir. Bu uygulama ile UKVA ortamında, hızlı portal geliştirimine ihtiyaç olduğu zaman, yeniden kullanılabilir portletlerle ile nasıl portal geliştirileceği bir portal uygulaması ile açıklanmıştır. Bu çalışmada UKVA Konumsal portalları için üç farklı portlet geliştirilmiştir. Bu portletler, “Harita Görüntüleyici” portleti, “Arama” portleti, “Metadata Editörü” portletidir (Sekil 19). “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 Geoportalında “Metadata Creator” web uygulamasındaki gibi veri sağlayıcılarının ISO19115 metadata standardına göre metadata üretmesi için kullanılacaktır. Bu portletler, JSR 286 standardına göre geliştirilmiştir. Bu portletlerin yanısıra, Liferay açık kaynak kodlu portal sunucusunun kullanıcılara sunduğu kullanıcı girişi portleti de bu uygulamada kullanılmıştır.



Şekil 18. Harita Görüntüleyici Portleti

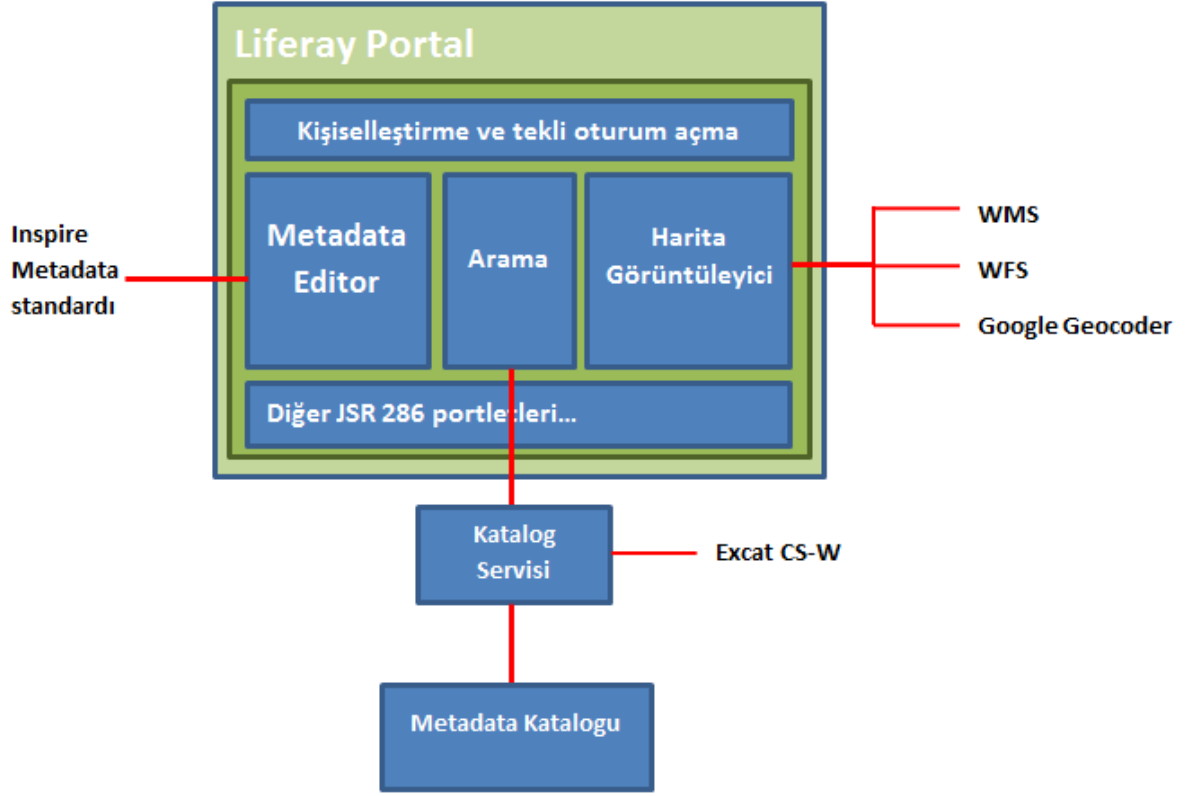


Şekil 19. Arama Portleti

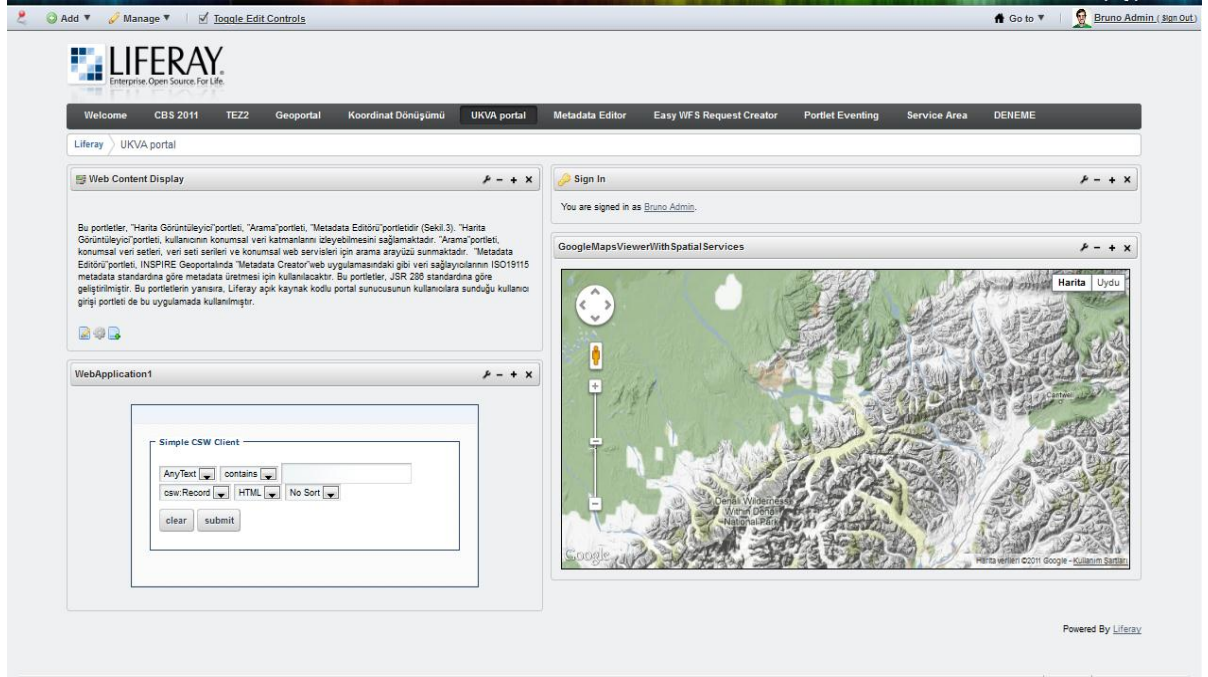


Şekil 20. Metadata Editör Portleti

Harita Görüntüleyici portleti, Google Maps API kullanılarak geliştirilmiştir. JSR-286 standardına uyumludur. Arama portleti Excat CSW sunucusunu kullanarak arama yapmaktadır. Metadata Editör portleti ise EUOSME projesi kapsamında, INSPIRE konumsal portalı için geliştirilen Metadata Creator uygulamasının portlet ortamına aktarılmasıyla geliştirilmiştir.



Şekil 21. Yeniden Kullanılabilir Portletlerle Geliştirilen Prototip Portal Mimarisi (Ball, 2005)' ten uyarlanmıştır.



Şekil 22. Çalışma Kapsamında Geliştirilmiş prototip Konumsal Portalın ekran görüntüsü

JSR-286 standartlarına göre geliştirilen “*Harita Görüntüleyici*” ve “*Arama*” portletleri, UKVA konumsal portalından, WSRP olarak yayınlanmıştır.

Bu sayede, son kullanıcı ihtiyaç duyduğu konumsal web servislerine ulaşmak için UKVA portalında arama yapmak yerine, WSRP olarak yayınlanan Arama portletine uzaktan erişim sağlayan başka bir portaldan UKVA konumsal portalında arama yapıyormuş gibi arama yapabilecektir. Bu çalışmada portal sunucusu olarak Liferay Portal sunucusu kullanılmıştır.

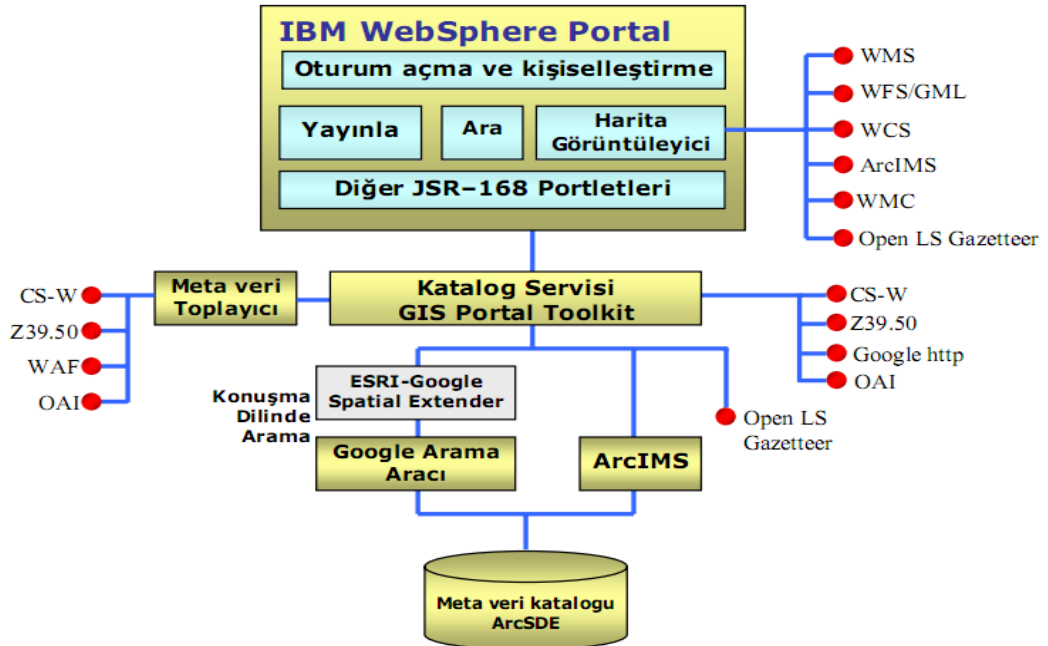
Konumsal portal geliştiriminin bu şekilde yapılması, geliştirilecek uygulamaların yeniden kullanılabilirliğini sağlamıştır. Bu çalışma kapsamında geliştirilen bu üç portlet, Liferay portal sunucusundan farklı bu standartları sağlayan herhangi bir portal sunucusunda kullanılabilir. Farklı kurumların farklı portal sunucuları kullanması bu uygulamaların bu portallarda kullanılmasını engellemeyecektir. 2.1 numaralı başlık altında ortaya koyulan çerçeve ile birlikte düşünüldüğünde, UKVA ortamında hızlı konumsal portal geliştirimine büyük katkıları olacaktır.

Geliştirilen konumsal portal, Foust vd., 2005’ te ki konumsal portal sınıflandırılmasına göre, katalog konumsal portal sınıfına girmektedir. Katalog konumsal portal mimarisi, dünya genelindeki UKVA çalışmalarında, çoğunlukla kullanılan yöntemdir. A.B.D. deki UKVA çalışmalarının gelişimi incelendiğinde, ilk etapta Clearinghouse yaklaşımı gerçekleştirilmiş daha sonra konumsal portal mimarisine geçiş yapılmış ve GOS portal geliştirilmiştir (Şekil 23).



Şekil 23. GOS portaldan ekran görüntüsü

GOS Portal mimarisinde (Şekil 24), merkezi bir metaveri kataloğundan konumsal veriler üzerinde arama yapılmaktadır. İhtiyaç duyulan veri bu arama sonucunda bulunduktan sonra uygulamanın gerçekleştirileceği ortama indirilmekte ve uygulama herhangi bir CBS yazılımı tarafından yapılmaktadır.



Şekil 24. GOS portal mimarisi (Ball, 2005)

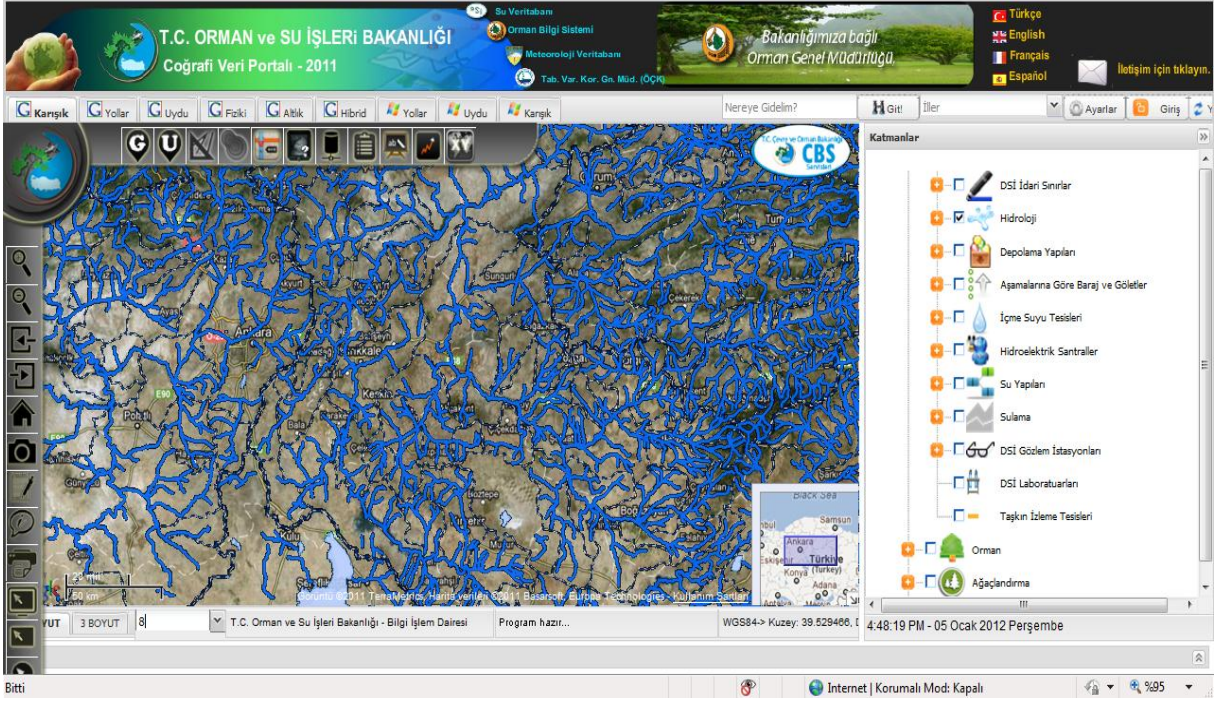
Dünya genelinde UKVA kapsamında geliştirilen portalların büyük çoğunluğunun metadata portalları olduğu görülmektedir. Çalışmanın bu kısmında gerçekleştirilen konumsal portal, GOS portal ile aynı mimariye sahiptir. Geliştirilen portalın, GOS portala göre üstünlükleri mevcuttur. İlk olarak, bu çalışma kapsamında geliştirilen konumsal portal, JSR-286 standardına uyumlu geliştirilmiştir. GOS portalı, JSR-168 standardına uyumludur. Literatürde, JSR-168 ve JSR-286 standartlarını karşılaştıran çalışmalar mevcuttur (Weiss, 2009). Geliştirilen konumsal portaldaki portletler, “Portlet Eventing” mekanizması ile birbiriyle iletişime geçebilmektedir. GOS portalda uygulamalar arasında herhangi bir iletişim söz konusu değildir. Her bir portlet uygulaması, portal ortamında kendi başına bir uygulamadır. Bu çalışmada, portletler arasındaki iletişimin sağlanabileceği gösterilmiş ve daha interaktif bir ortam oluşturulmuştur.

Ülkemizde, UKVA çalışmalarında olduğu gibi konumsal portal çalışmalarında da geç kalınmıştır. Dünyada, bir çok ülke konumsal portalını geliştirmiş ve süreç içerisinde portal uygulamasında geliştirmeler yapmaktadır. Ülkemizde henüz ulusal düzeyde, UKVA konumsal portalı geliştirimi yapılamamıştır. Kurum düzeyinde kurumsal portal çalışmaları mevcuttur. Fakat bu çalışmalar da yeterli düzeyde değildir. T.C. Orman ve Su İşleri Bakanlığı tarafından konumsal portal çalışması yapılmıştır (Şekil 25). Bu konumsal portal için Esri Geoportal Server yazılımı kullanılmıştır. Katalog konumsal portal sınıfına giren bir konumsal portal uygulamasıdır.



Şekil 25. T.C. Orman ve Su İşleri Bakanlığı bünyesinde gerçekleştirilen konumsal portal uygulaması

Bu konumsal portalın çalışma performansı ve sunduğu hizmetler göz önüne alındığında, servis kalitesinin alt düzeyde olduğu görülmüştür. Örneğin; arama sonucunda bulunan verinin, harita görüntüleyici Şekil 26'da görülmektedir. Bu harita görüntüleyeci her ne kadar Coğrafi Veri Portalı olarak adlandırılrsa da, bu çeşitli katmanların web üzerinden sunulduğu bir harita görüntüleme arayüzüdür. Yüksek hızlı internet ortamında çalışılmasına rağmen, bu harita görüntüleme arayüzünün performansı son derece düşüktür.



Şekil 26. Harita Görüntüleme Arayüzü

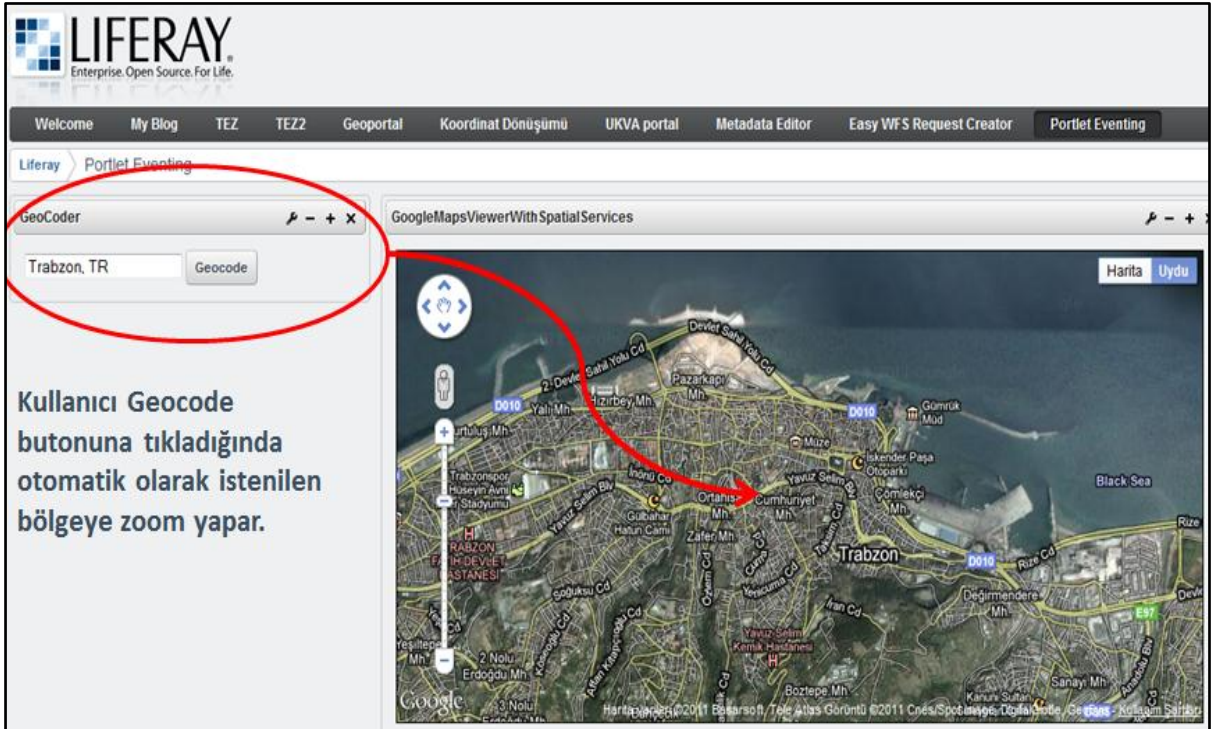
Yapılan son incelemelerde ise, Coğrafi Veri portalının aktif olduğu gözlemlenmiştir. Ülkemizde ulusal düzeyde konumsal veri ile iş yapan kesimler arasında “birlikte işlerlik” sağlayacak bir konumsal portal geliştirimi gerçekleştirilememiştir. UKVA çalışmalarında geride kalmış olan ülkemizin bir an önce konumsal portal geliştirimini gerçekleştirmesi gerekmektedir. Dolayısıyla, ülkemiz hızlı portal geliştirimine ihtiyaç duymaktadır. Bu tez kapsamında ortaya koyulan hızlı portal geliştirimi yaklaşımı, bu bakımdan önem arz etmektedir. Bu yaklaşım, ülkemizde olduğu gibi dünya genelinde UKVA çalışmalarında geride kalmış ülkeler tarafından uygulanabilecek bir yaklaşımdır.

2.3 Portal Teknolojisi ve Web Servisi Kompozisyonu

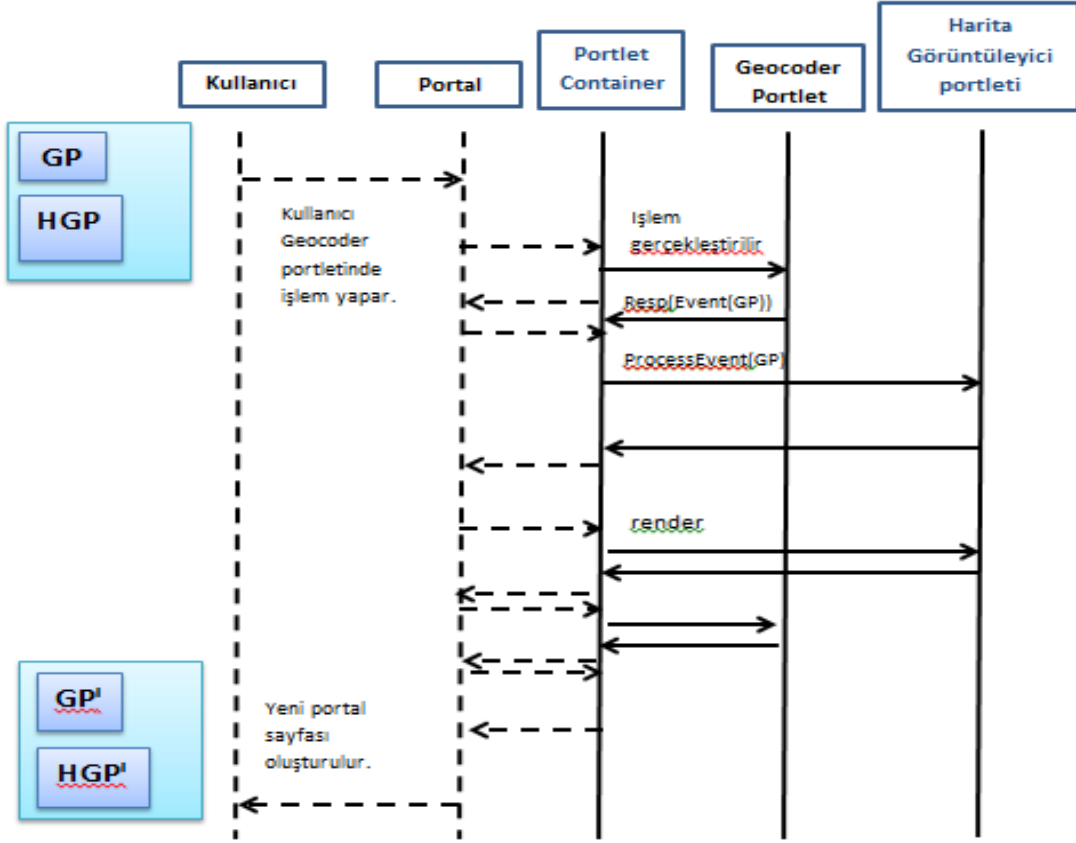
WSK, bir kullanıcı tarafından gerçekleştirilmek istenen karmaşık bir uygulamanın, tek bir Web servisi tarafından yerine getirilememesi durumunda, uygun Web servislerinin birleştirilerek uygulamanın gerçekleştirilmesi şeklinde tanımlanabilir (Akıncı, 2006). WSK, Web servislerinin ya da işlemlerin bir işbirliğine olanak sağlayacak tarzda birbirleri ile ilişkilendirilmesidir (Peltz, 2003). WSK, SYM'nin asıl katkı noktalarından bir tanesi

olduğu için önem arz etmektedir. Bu çalışmanın amaçlarından bir tanesi olan, portal ortamında web servisleri kompozisyonun gerçekleştirilmesi, bu bakımdan önemlidir.

Bu çalışma kapsamında, konumsal portallar 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. Portlet Eventing mekanizması, birinci bölümde 1.8 başlığı adı altında açıklanmıştır. Portal ortamında WSK’ ya örnek olarak basit bir uygulama geliştirilmiştir. Bu uygulama, iki web servisi ve iki portlet den oluşmaktadır. Bunlar, “Geocode” ve “Google Maps” servisi 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 27).



Şekil 27. Portlet Event ile web servisi kompozisyonuna basit düzeyde bir örnek



Şekil 28. Örnek WSK iş akışı UML diagramı

Birinci bölümde 1.2 başlığı adı altında anlatıldığı gibi, katalog ve uygulama tabanlı konumsal portallar mevcuttur. İkinci bölümde, 2.2 başlığı adı altında anlatılan, çalışma kapsamında geliştirilen prototip konumsal portal, katalog konumsal portaldır. Çünkü bu portal merkezi metaveri kataloğuna sahiptir. Kullanıcı, bu metaveri üzerinden ihtiyaç duyduğu konumsal veriseti/ veriseti serileri ve web servisleri için arama yapmaktadır.

Fakat bu yapı günümüzün ihtiyaçlarını karşılayamamaktadır. Dolayısıyla, katalog portallarından, uygulama tabanlı portallara geçiş yapılması artık zorunlu hale gelmiştir. Uygulama tabanlı portaldan kasıt; kullanıcıların konumsal veriyle ilgili uygulamalarını portal ortamında gerçekleştirmesi kastedilmektedir.

2.4. WPS ile Web Tabanlı Konumsal Analiz

Web 2.0 teknolojilerinin hızla geliştirilmesiyle, web haritacılık teknolojileri de hızla gelişmiştir. Web haritacılık teknolojilerinin gelişmesiyle, web üzerinden konumsal verinin

sunumunun yanısıra, web üzerinden konumsal veri üzerinde CBS analizlerinin gerçekleştirilmesi ihtiyacı ortaya çıkmıştır.

1.7 başlığı adı altında ayrıntılı bir şekilde anlatılan WPS, bu bakımdan önem arz etmektedir. Web ortamındaki bu ihtiyacın giderilmesine yönelik OGC tarafından WPS standardı tanımlanmıştır. WPS standardının ilk sürümünün 2005 yılında yayınlanmış olması göz önüne alındığında, WPS yaklaşımının, son bir kaç yıldır popülerlik kazandığı görülmektedir. WPS yaklaşımının son bir kaç yıldır popülerlik kazanmasının sebeplerinden en önemlisi, web üzerinden konumsal analiz ihtiyacının, son zamanlarda ortaya çıkması olarak gösterilebilir. Dünya genelindeki UKVA çalışmalarında, ilk etapta konumsal verilerin web üzerinden sunulmasına yönelik uygulamaların geliştirilmesi nedeniyle, WFS, WMS, WCS gibi OGC web servisleri popülerlik kazanmıştır. Bu süreç tamamlandıktan sonra,web ortamında konumsal analiz ihtiyacı ortaya çıkmıştır. Bu nedenle, WMS ve WFS gibi servisler için geliştirilen sunucuların sayısı göz önüne alındığında, WPS servisi için geliştirilen sunucu sayısının az olduğu görülmektedir. Deegree WPS, Geoserver WPS, 52 North WPS, PyWPS, Zoo WPS sunucuları, geliştirilen WPS sunucularıdır.

Bu çalışma kapsamında, literatür taraması yapılarak, WPS alanındaki son durum ve gerçekleştirilen uygulamalar incelenmiş, WPS yaklaşımının, konumsal portal bağlamında sunabileceği katkı belirlenmeye çalışılmıştır. Bu kaynaklardan bazıları, (Granell vd., 2007), (Meng vd., 2010), (Chung vd., 2009), (Kiehle vd. 2007), (Fleuren ve Müller, 2008), (Stollberg ve Zipf, 2009), (Stollberg ve Zipf, 2007), (Lanig vd., 2011), (Brauner vd., 2009), (Arindam ve Ghosh, 2010), (Stollberg ve Zipf, 2009) dur.

Literatürde, WPS yaklaşımı birçok alanda kullanılmıştır. WPS, deprem afet değerlendirme sisteminin oluşturulmasında (Meng vd.,2010), emlak piyasasında kullanıcıların kendileri için uygun emlak bulabilmesi için yer seçimi analizinin gerçekleştirilmesinde (Stollberg ve Zipf, 2009), çeşitli hidrolojik modellemelerin gerçekleştirilmesi ile sel, taşkın vb. afetlerin vereceği zararın öngörülmesi ve zararın minimuma indirilmesi için (Granell vd., 2007), rüzgar enerji santrallerinin yer seçimi analizinde (Lanig vd., 2011), insanların yoğun olarak yaşadığı bölgelerde bomba patlaması sonucu, zarar görebilecek alanların tespiti ve hatta üç boyutlu bir şekilde hangi binaların kaçınıcı katına ne kadar etki yapacak bunun belirlenmesinde (Stollberg ve Zipf, 2007)

kullanılmıştır. Görüldüğü üzere, WPS kullanım alanları çeşitlilik göstermektedir. Bu uygulamalar, WPS sunucu ortamında gerçekleştirilmiş uygulamalardır. İlerleyen başlık altında irdelenecek olan bir çok eksiklikleri mevcuttur. Bu çalışma kapsamında, literatürde yapılan çalışmalardan farklı olarak, WPS ve portal teknolojisinin entegrasyonunun sağlanmasıyla, CBS analizlerin konumsal portal ortamında gerçekleştirilebileceği bir konumsal portal ortamı yaratmaktır.

Tez kapsamında kullanılacak WPS sunucusunun belirlenmesi, literatürde WPS sunucularına ait dokümanların incelenmesi (Deegree WPS, 2009), (Geoserver WPS, 2011), (52 North WPS, 2011), (PyWPS, 2010), (Zoo WPS, 2011), literatürde mevcut olan WPS sunucularını inceleyen ve birbirine göre üstünlüklerini araştıran çalışmanın incelenmesi (Schaffer, 2010) ve tez kapsamında bütün bu sunucuların kurulmasının yapılması ve incelenmesiyle belirlenmiştir. WPS sunucuları, kullanım ve WPS işlemi geliştirme kolaylığı, kullanıcı arayüzü ve standarda uyumluluğu gibi parametrelere göre irdelenmiştir. Tablo 8’ de (Schaffer, 2010) çalışmasında OGC WPS standardına göre uyumluluk testi sonuçları gösterilmektedir.

Tablo 8. WPS sunucularının OGC WPS standardına göre uyumluluk testi

Sonuç/ WPS sunucuları	52 North	Deegree WPS	pyWPS	ZooWPS
GetCapabilities isteği	✓	✓	×	×
DescribeProcess isteği	✓	✓	×	×
Execute İsteği	✓	✓	×	×

Yapılan çalışmalar sonucunda öne çıkan WPS sunucuları, 52 North WPS ve Deegree WPS sunucuları olarak belirlenmiştir. 52 North WPS sunucusu kullanıcılara, toplamda 395 adet WPS işlemi sunmaktadır. Deegree WPS sunucusu ise 11 adet Java Topology Suite (JTS) WPS işlemi sunmaktadır. Bu çalışma kapsamında varolan WPS işlemlerinin kullanılması yerine, yeni WPS işlemleri geliştirilecektir. Dolayısıyla, bu iki sunucudan birisinin seçilmesindeki, en önemli kriter, yeni WPS işlemi geliştiriminin kolaylığı ve kullanıcı arayüzünün basitliğidir. Sonuç olarak, Deegree WPS, yeni WPS işlemi

geliştiriminin 52 North WPS sunucusuna göre kolay olması sebebiyle seçilmiştir. Bu çalışma kapsamında, genel olarak Deegree WPS sunucusu kullanılmıştır. Sadece, 2.5 başlığı altında anlatılacak olan “Kvp İsteklerle Az Sayıda Servisin Kompozisyonu” konusunda Geoserver WPS kullanılmıştır. Bunun sebebi; Deegree WPS sunucusunun KVP encoded istekler Execute operasyonunun gerçekleştirilmesini desteklememesidir.

Bu çalışma kapsamında, katı atık depolama tesisi yapılamayacak alanların tespitinin, portal ortamında, WPS ile gerçekleştirilmesi amaçlanmaktadır. Bu analizin gerçekleştirilmesi için yeni WPS işlemlerin geliştirilmesi gerekmektedir. Bu tez kapsamında gerçekleştirilecek uygulamalar için yeni WPS işlemleri geliştirilmiştir. Tablo 9 ‘da bu çalışma kapsamında geliştirilen WPS işlemleri görülmektedir.

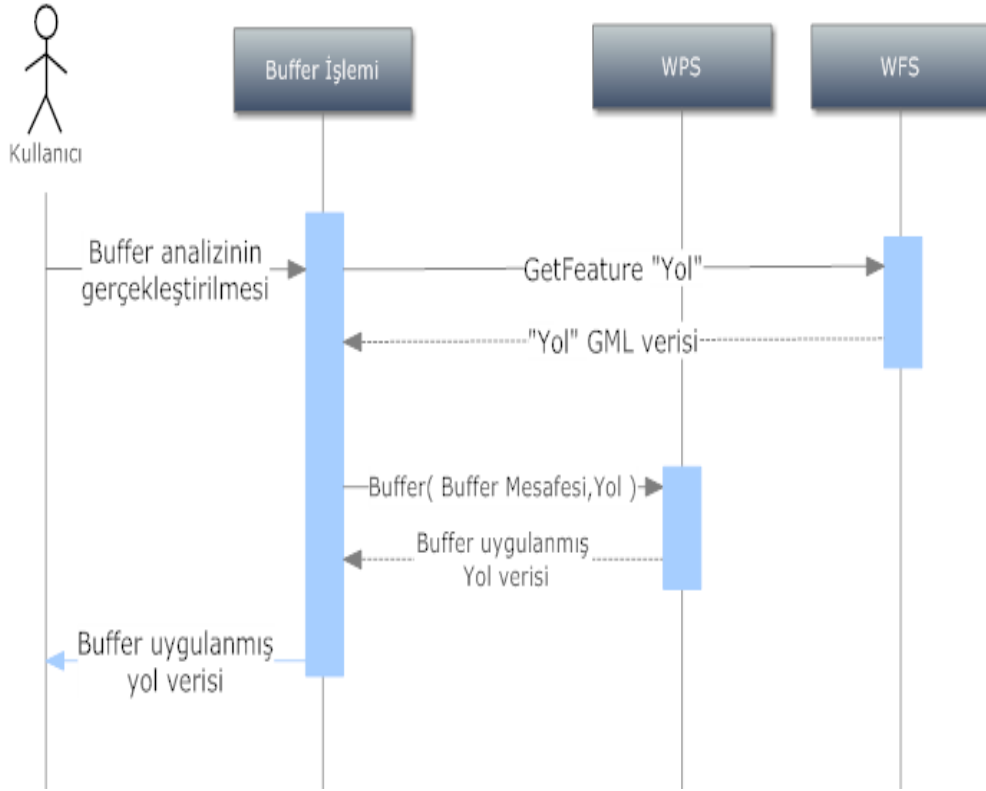
Tablo 9. Tez çalışması kapsamında geliştirilen WPS işlemleri

Geliştirilen WPS işleminin tanımlayıcısı	Gerçekleştirdiği işlem
bufferMEY	Deegree WPS sunucusu tarafından sağlanan, buffer işlemi sadece GML 3.1.1 versiyonu ile çalışabilmektedir. Geliştirilen bufferMEY işlemi ile GML’in 3.1.1, 2.1.2, 3.0.1, 3.2.1 versiyonlarına buffer uygulanabilmektedir. Ayrıca, geliştirilen bu işlem ile bir WFS sunucusu tarafından gelen GML verisine buffer analizi gerçekleştirilebilmektedir.
intersectMEY	Deegree WPS sunucusu tarafından sağlanan, intersect işlemi sadece GML 3.1.1 versiyonu ile çalışabilmektedir. Geliştirilen intersectMEY işlemi ile GML’in 3.1.1, 2.1.2, 3.0.1, 3.2.1 versiyonları ile intersect analizi gerçekleştirilebilmektedir. Ayrıca, geliştirilen bu işlem ile bir WFS sunucusu tarafından gelen GML verileri ile intersect analizi gerçekleştirilebilmektedir.
unSuitZoneMEY	Geliştirilen bu WPS işlemi, katı atık depolama tesisi yapılamayacak alanların tespitini gerçekleştirmektedir. GML’in 3.1.1, 2.1.2, 3.0.1, 3.2.1 versiyonları ile çalışabilmektedir. Bir WFS sunucusundan gelen GML verisi ile işlem yapabilmektedir.

Tablo 9'un devamı

bufferIntRoadMEY	Geliştirilen bu WPS işlemi, kompozit WPS işlemine örnek olması bakımından geliştirilmiştir. Bu WPS işlemi, ilk önce yol verisine buffer uygulamakta ve sonuç katmanı ile kadastral katman kullanılarak intersect analizini gerçekleştirerek kamulaştırma yapılacak parsellerin tespiti işlemini gerçekleştirmektedir.
------------------	---

Geliştirilmiş WPS işlemleri, Deegree WPS ortamında çalışmaktadır. Bu WPS işlemleri WFS sunucusu tarafından sunulan GML 3.1.1, 2.1.2, 3.0.1, 3.2.1 veriler ile çalışabilmektedir. Uygulamanın gerçekleştirilmesinde kullanılacak tüm veriler, Geoserver sunucusu kullanılarak WFS ile sunulması planlanmıştır. WPS işlemleri, bu yaklaşıma göre geliştirilmiştir.



Şekil 29. Buffer WPS İşleminin iş akışı

2.5 WPS ile WSK

Konumsal portal kullanıcıları, konumsal analizleri gerçekleştiren web servislerine ve bu web servislerinin zincirlenmesi ile kompleks CBS analizlerin gerçekleştirilebildiği bir portal ortamında ihtiyaç duymaktadır. Konumsal analizlerin, web servisleri tarafından gerçekleştirilmesi için, The Open Geospatial Consortium (OGC) tarafından, 2005 yılında Web Processing Service v0.4.0 standardı yayınlandı (OGC, 2005) ve 2007 yılında güncellenerek Web Processing Service v1.0.0 (WPS) standardı yayınlanmıştır (OGC, 2007). Bu servis sayesinde web üzerinden herhangi bir CBS konumsal analiz fonksiyonu kullanılabilir ya da aynı bölgeye ait iki raster görüntüyü karşılaştırılıp değişim analizini gerçekleştirilebilir ya da vektör veri üzerinde genelleştirme işlemleri uygulanabilir. Bu çalışma kapsamında, WPS standardını sağlayan açık kaynak kodlu yazılımlar incelenmiştir. Bunlar; 52 North WPS, Deegree WPS, Geoserver WPS, pyWPS'dir. WPS servislerinin kompozisyonu ile, katı atık yer seçimi gibi kompleks bir CBS analizi gerçekleştirilebilmektedir.

Bu çalışma kapsamında, uygulama tabanlı portala örnek teşkil edecek bir uygulama tabanlı portalda geliştirilmiştir. Portlet Eventing mekanizması, portal ortamında kompleks konumsal analizlerin gerçekleştirilmesinde kullanılacaktır. Portal ortamında konumsal analizlerin gerçekleştirilmesinde, birinci bölümde 1.7 başlığı adı altında anlatılan Web Processing Servisi kullanılacaktır.

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 kalınmıştır. 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?

2.5.1 Koşum Zamanında Servis Kompozisyonu

2.5.1.1 KVP Kodlanmış 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. 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.

WPS sunucusuna gönderilen yol verisine uygulanacak buffer isteği:

```

http://localhost:6161/geoserver/ows?service=wps&

version=1.0.0&

request=Execute&

Identifier=JTS: Buffer&

DataInputs=features=@mimeType=text/xml@href=http://localhost:/geoserver/tiger/ows?
service=WFS&version=1.0.0&request=GetFeature&typeName=tiger:roads@Schema=htt
p://localhost:6161/geoserver/schemas/wfs/1.1.0/wfs.xsd&

RawDataOutput=bufferedGML@mimeType=text/xml

```

WFS sunucusuna Kadastro parseli için gönderilen istek:

```

http://localhost:6161/geoserver/tiger/ows?service=WFS&

version=1.0.0&

request=GetFeature&

typeName=tiger:CadastreParcels@Schema=http://localhost:6161/geoserver/schemas/wfs
/1.1.0/wfs.xsd

```

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

```

http://localhost:6161/geoserver/ows?service=wps&

version=1.0.0&

request=Execute&

Identifier=JTS: Intersect&

DataInputs=features=@mimeType=text/xml@href=http://localhost:6161/geoserver/tiger/
ows?service=WFS&version=1.0.0&request=GetFeature&typeName=tiger:CadastreParc
els@Schema=http://localhost:6161/geoserver/schemas/wfs/1.1.0/wfs.xsd,

```

```
http://localhost:6161/geoserver/ows?service=wps&version=1.0.0&request=Execute&Identifier=JTS:Buffer&DataInputs=features=@mimeType=text/xml@href=http://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=bufferedGML@mimeType=text/xml&
```

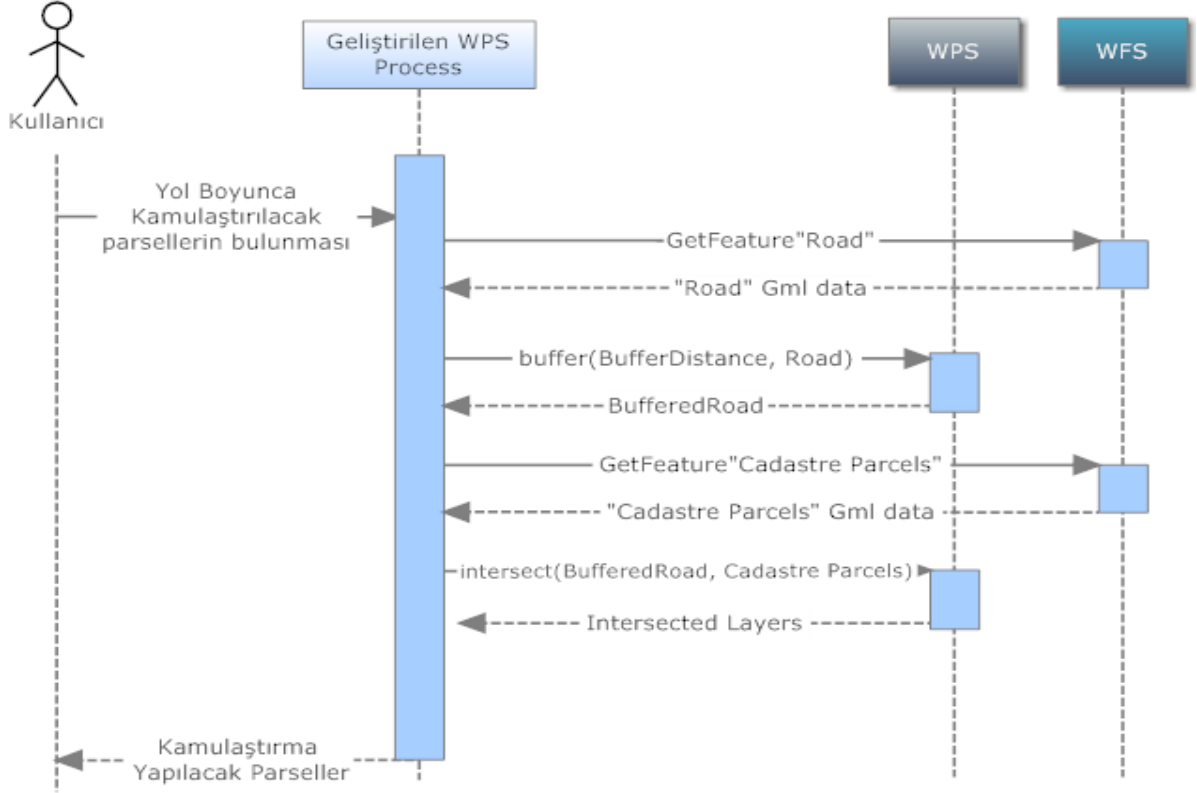
```
RawDataOutput=intersectoutput@mimeType=text/xml
```

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.

2.5.2 Tasarım Zamanında Servis Kompozisyonu

2.5.2.1 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 (Beate vd, 2007), (Beate 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 30). Çalışmada, sadece WPS işlemleri birbiriyle kompoze 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 (Beate vd, 2009).



Şekil 30. Geliştirilen WPS processinde 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 olacaktır. Geliştirilen işlem, sadece analizin gerçekleştirilmesi için kullanılabilir olacaktır. UKVA gibi dağıtık sistemlerde, farklı geliştiricilerin geliştirdiği servislerin kompozisyonu bu şekilde sağlanamayacaktır.

2.5.2.2 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 dökümanı da bu çalışmada üretilmiştir. Ayrıca bu çalışmada Deegree WPS te tanımlı "buffer" ve "intersect" işlemlerinden farklı² "buffer" ve "intersect" işlemleri tanımlanmıştır. Aşağıdaki WSDL dökümanı, bu çalışma kapsamında üretilen WSDL dokümanının "mesajlar" ve "port tipi" kısmından bir parçadır.

```
<wsdl:message name="DescribeProcess_BufferSoapRequest">
<wsdl:part name="parameters" element="tns:DescribeProcess_Buffer"/>
</wsdl:message>
<wsdl:message name="DescribeProcess_BufferSoapResponse">
<wsdl:partname="parameters" element="tns:DescribeProcess_BufferResponse"/>
</wsdl:message>
...
<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"/>
```

² Farklı girdi ve çıktı formatları ile.

```

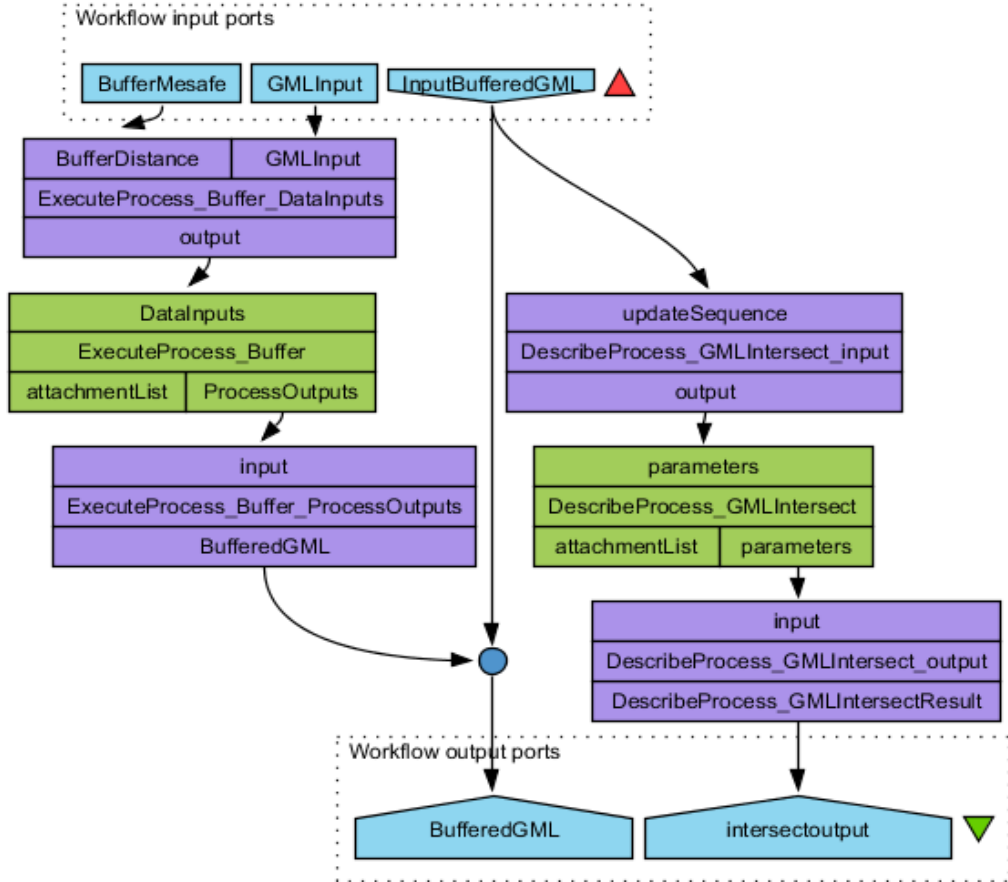
<output message="tns:ExecuteProcess_BufferResponse"/>
<fault name="ExceptionResponse" message="tns:ExceptionResponse"/>
</operation>
...
</portType>

```

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 sunmalıdır. 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/identifier [/service.soap]?WSDL” şeklinde tüm işlemler için ayrı ayrı sunmaları tavsiye edilmiştir. Bu çalışma kapsamında incelenen 52 North WPS, Degree WPS, Geoserver WPS, pyWPS sunucuları, standartta dile getirilen WSDL dokümanın 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 (identifier) göre ayrı ayrı WSDL tanımlarını gerçekleştiren WPS sunucusu mevcut değildir. Dolayısıyla, kullanıcı tüm WPS işlemlerini kapsayan büyük bir WSDL dokümanı ile çalışmak zorunda kalmaktadır. Ayrıca, her işlem için ayrı ayrı WSDL dokümanının oluşturulması, servis kataloglarına servisteki tüm işlemlerin tamamının kaydolması yerine işlemler ayrı ayrı kaydolabilecektir. 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 kompoze 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 31). Oluşturulan bu iş akışı, web üzerinden çağrılabilir ve başka iş akış şemalarının içerisinde kullanılabilir. Literatürde, WPS işlemlerinin bu yazılım ile kompozisyonunun yapılmasına yönelik çalışma mevcuttur (De Jesus, vd., 2011).



Şekil 31. Taverna Workbench ile oluşturulan iş akış diagramı

2.6 Geliştirilen Uygulama Tabanlı Konumsal Portalların İrdelenmesi ve Prototip Konumsal Portal Geliştirimi

(Foust vd., 2005) uygulama tabanlı portalı, katalog konumsal portalda gerçekleştirilebilen konumsal veri üzerinde arama, izleme ve indirme servislerinin yanısıra, özel bir ihtiyaç doğrultusunda geliştirilen arayüzü ile bu ihtiyaç doğrultusunda daha etkin hizmet vermeyi amaçlayan portallar olarak tanımlamaktadır. (Foust vd., 2005)'e göre, uygulama portalları, kullanıcılara uygulama gerçekleştirecekleri veri üzerinde, güzergah bulma, geo-coding, karmaşık sorgular, editleme ve güncelleme gibi hizmetleri sunmalıdır.

Uygulama tabanlı konumsal portalın bu tanımı, günümüzün ihtiyacını karşılayamamaktadır. Dünya genelindeki katalog konumsal portallar incelendiğinde, tanımda ortaya koyulan hizmetlerin büyük bir kısmını, kullanıcılara hizmet olarak sunulduğu görülmektedir. Bir konumsal portalın, uygulama tabanlı portal olabilmesi için portal kullanıcısının konumsal veri üzerinde çeşitli CBS uygulamalarını gerçekleştirebiliyor olması gerekmektedir. Basit konumsal analizlerin (buffer, intersect, clip vb.) yanı sıra daha karmaşık konumsal analizlerin (yer seçimi, ağ analizi vb.) gerçekleştirilebildiği bir konumsal portal ortamı, ancak uygulama konumsal portalı olarak tanımlanabilir. SYM mimarisinde, bir web servisi tarafından yerine getirelemeyecek uygulamalar, WSK ile gerçekleştirilmektedir. Etkin SYM ortamına geçiş sürecinde ara geçiş evresi olarak tanımlanan portal ortamında, karmaşık konumsal analizlerin gerçekleştirilebilmesi için, WSK'nın portal ortamında gerçekleştirilebilmesi gerekmektedir.

Literatürde, uygulama tabanlı konumsal portala yönelik örnek uygulamalar mevcuttur. AWARE³ ve GENESIS⁴ konumsal portal uygulamaları, bu konuda örnek uygulamalardır. Aware konumsal portalı, hidrografik veriler üzerinde, çeşitli konumsal analizlerin gerçekleştirilebilmesi ve sel, taşkın vb. afetlerde çevreye verilecek zararın tespiti ve önceden önlem alınabilmesine yönelik geliştirilmiş bir portaldır.

³ AWARE: A tool for monitoring and forecasting Available Water Resource in mountain environment <http://www.aware-eu.info>

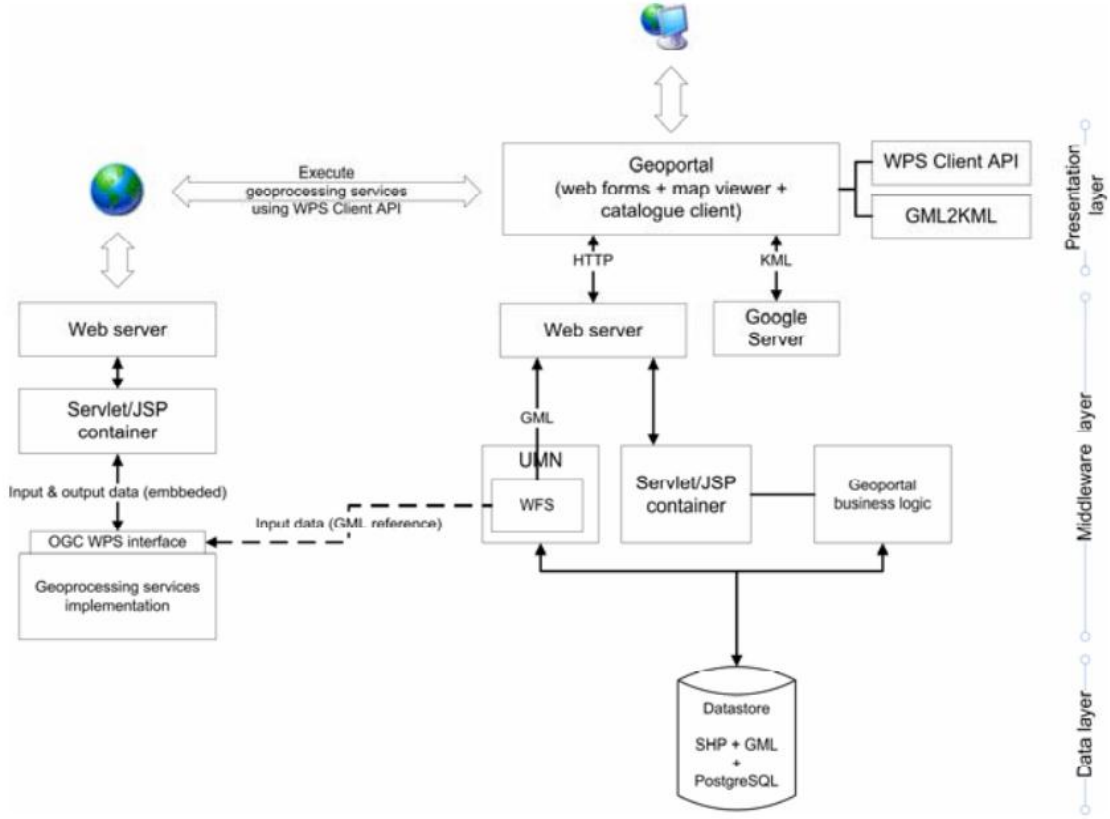
⁴ Genesis Geoportal, Generic European Sustainable Information Space for Environment, <http://www.genesis-fp7.eu/>

2.6.1. AWARE Konumsal Portal Mimarisi

AWARE konumsal portalı, 2005 yılında başlayan Avrupa Birliği tarafından desteklenen, 1.292.750 € proje bedeli olan, “*Available WAter REsource in Mountain Environment*” (AWARE) projesi kapsamında üretilmiştir. Bu konumsal portal uygulaması, belirli sorunlara etkin bir şekilde çözüm sunabilecek hidrografik akış modellerini konumsal portal ortamında, belirli kullanıcı gruplarına sunabilmeyi hedeflemektedir (Rampini vd., 2006). Bu proje kapsamında, hidrografik modellerin portal üzerinde gerçekleştirilmesine yönelik çalışmalar gerçekleştirilmiştir.

Bu portal, hidrolojik modellerin web üzerinden portal arayüzü ile gerçekleştirilmesi için, WPS ile web harita görüntüleme servislerinin entegrasyonu sağlanarak, kullanıcılara uygulamaları gerçekleştirilebildiği bir arayüz, sunmaktadır (Diaz vd., 2010). Bu konumsal portal, hidrografik modellerin gerçekleştirilebilmesi için gerekli konumsal veri/veriseti ve servislerin sunulduğu sunucu tarafı (server-side) ve kullanıcıların konumsal veri/veriseti ve servisler için arama yapabildikleri katalog arama arayüzü, hidrografik modellerin, çeşitli parametrelerin kullanıcıların tarafından belirlenmesiyle gerçekleştirilebilmesi ve analiz sonucunun Google Maps API⁵ ile geliştirilmesi harita arayüzünden sunulmasını sağlayan kullanıcı tarafı (client-side) olarak, iki kısma ayırmak mümkündür. (Diaz vd., 2010), AWARE portal mimarisini üç katmana ayırmıştır. Bu katmanlar; sunum katmanı, veri katmanı ve middleware katmanıdır (Şekil 32).

⁵ Web üzerinde google haritaları tabanlı web arayüzü geliştirimi için API, <http://code.google.com/intl/tr-TR/apis/maps/index.html>



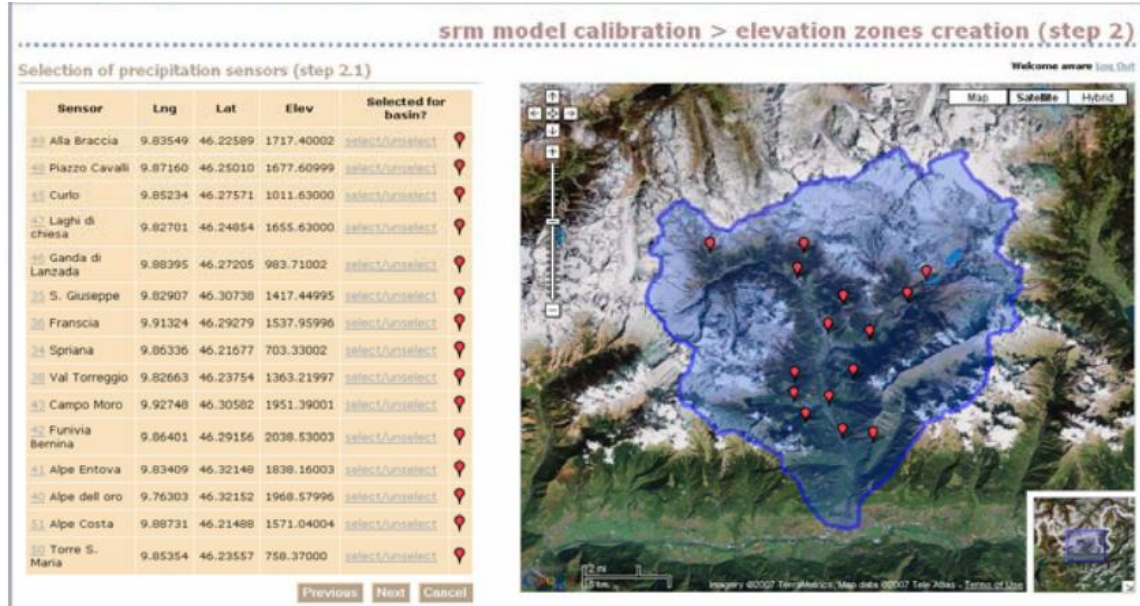
Şekil 32. AWARE Konumsal Portal Mimarisi (Diaz vd., 2010)

Sunum katmanı: Hidrografik modellerin parametrelerinin web üzerinden konunun uzmanları tarafından belirlenmesini ve bu modelin gerçekleştirilmesini sağlayan, konumsal portalın kullanıcı arayüzüdür. Kullanıcıların, konumsal portaldaki uygulamaları, kullanmak için portala kayıt olması, gerekmektedir. Üyelik bilgilerinin saklanması ve portala giriş esnasında sorgulanabilmesi için, açık kaynak kodlu veritabanı yazılımı olan PostgreSQL kullanılmıştır. Portal arayüzü, Apache Struts⁶ kullanılarak geliştirilmiştir.

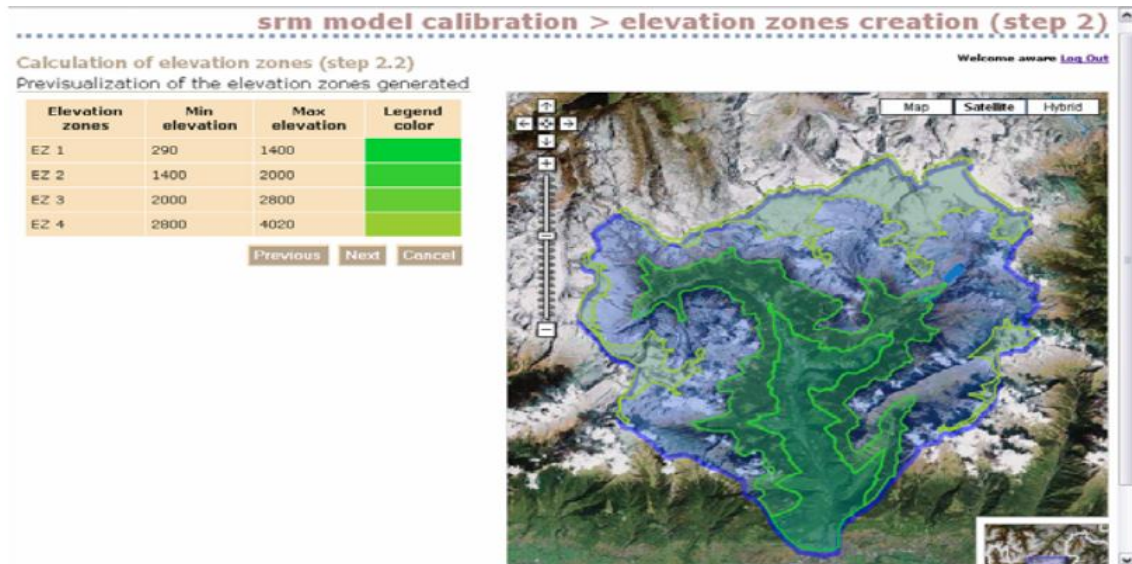
Middleware katmanı: hidrografik modellerin gerçekleştirimi için gerekli tüm web servislerinin olduğu katmandır. Konumsal analizlerin gerçekleştirilmesi için kullanılan bir WPS sunucusu, konumsal verinin sunulması için bir WFS sunucusu, portalın koşturulduğu bir web sunucusu ve Google Haritalarını sunan Google sunucusundan oluşmaktadır. Bu portal uygulamasında, OGC WPS 0.4.0 standardı kullanılmış ve 52 Norht WPS sunucusu kullanılmıştır. Konumsal analizler, bu servis vasıtasıyla gerçekleştirilmektedir.

⁶ Apache Struts, dinamik java web uygulamaları geliştirmek için bir çatı, <http://struts.apache.org/>

Portal, Apache Tomcat sunucusunda çalışmaktadır. Portal üzerinde analiz sonucunun gösterimi için Google Maps API kullanılarak geliştirilmiş, harita görüntüleyici kullanılmıştır.



Şekil 33. Yağış miktarı ölçen istasyonların dağılımı ve havza sınırını gösteren portal ekran görüntüsü (Granell vd., 2008)



Şekil 34. Havza sınırı içerisinde yükseklik bölgelerini gösteren portal ekran görüntüsü (Granell vd., 2008)

Aware konumsal portalında, 52 North WPS sunucusu tarafından sunulan WPS işlemleri ve proje grubu tarafından geliştirilmiş WPS işlemleri kullanılmıştır (Tablo 10).

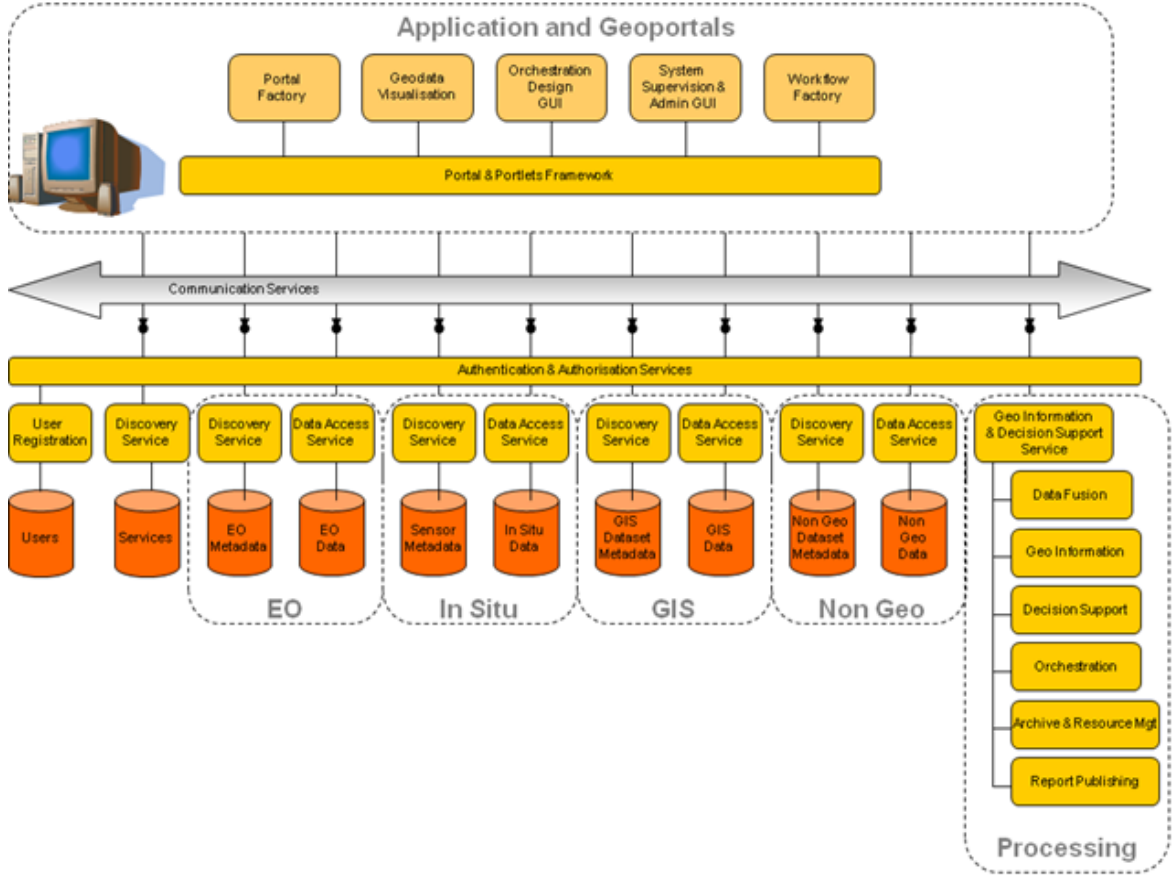
Tablo 10. AWARE konumsal portalında kullanılan WPS işlemleri

WPS process grubu	WPS işlemleri
Topoloji WPS	Area, Intersect, Union, Contains, Buffer, MaxExtent
Görüntü işleme WPS	Slicing, SlicingRanges, Vectorize, CoordinateElevation, StationsElevation, HypsometricElevation, ElevationZoneCalculation, SnowCoverAreaCalculation
Grafik WPS	DepletionCurvesPlot, DischargePlot
Koordinat dönüşüm WPS	ransCoordGMLPoint, TransCoordPoint, TransCoordPoint7P
Veri Format WPS	TransSHPEPSG, TransSHPtoGML

Aware konumsal portalındaki hidrografik modellerin gerçekleştirilmesinde kullanılan WPS işlemleri Tablo 10' da ki gibidir (Diaz vd., 2010). Konumsal portal ortamında, WPS servisi vasıtasıyla konumsal analizlerin gerçekleştirilebildiği bir ortamın ilk örneklerinden bir tanesi olması bakımından, önemli bir uygulamadır.

2.6.2. GENESIS Konumsal Portal Mimarisi

Konumsal portal ortamında, konumsal analizlerin gerçekleştirilmesine yönelik bir diğer örnek uygulama ise, GENESIS konumsal portalıdır. GENESIS projesi, 2008 ve 2011 yılları arasında AB tarafından desteklenen bir projedir (GENESIS, 2011). Bu projede, AWARE konumsal portalında olduğu gibi herhangi bir tematik konuda çözüm sunmak yerine, bu şekilde herhangi bir çevresel sorunun çözümüne yönelik portal ortamı ihtiyacı doğduğunda, bunun GENESIS portal mimarisi ve bu kapsamda geliştirilmiş bileşenlerle gerçekleştirilmesi amaçlanmaktadır. Bu doğrultuda GENESIS konumsal portal mimarisi kullanılarak, hava ve su kalitesinin analizine yönelik pilot çalışmalar, gerçekleştirmiştir.

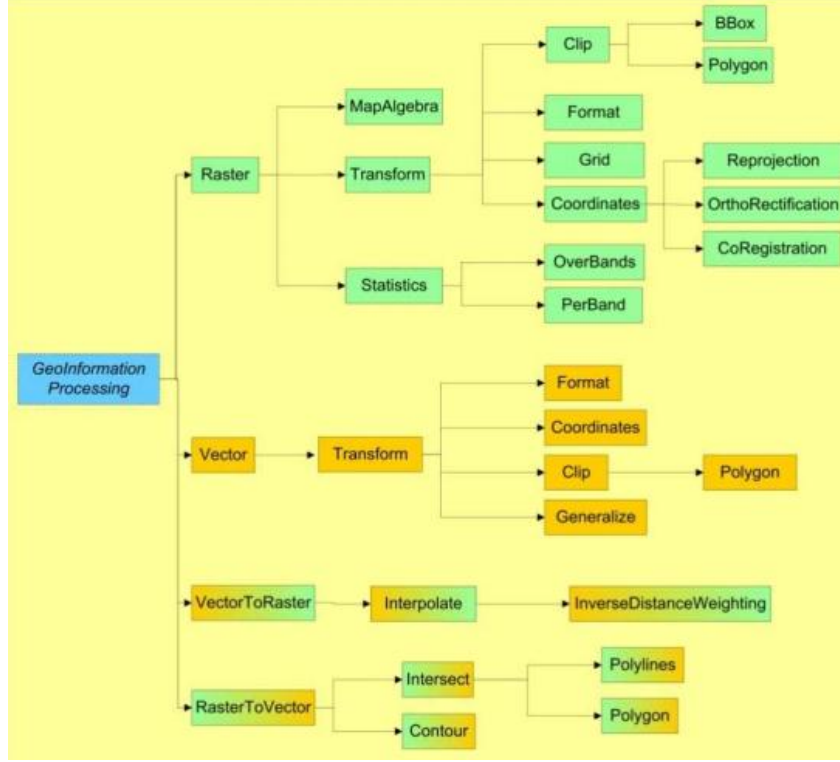


Şekil 35. GENESIS konumsal portal mimarisi

GENESIS portal mimarisi, sunucu tarafı ve kullanıcı tarafı olmak üzere iki kısımda incelenebilir. Kullanıcı tarafında, konumsal portalın kullanıcı arayüzünü oluşturana portal ve portletler bulunmaktadır. Kullanıcı, bu arayüzden ihtiyaç duyduğu analizi gerçekleştirecektir. Sunucu tarafında ise, uygulamanın gerçekleştirileceği servisler bulunmaktadır. Bu servisler, konumsal veri üzerinde metadata standartlarına göre arama yapabilmeyi sağlayan katalog servisler (discovery service), konumsal verinin web üzerinden sunumuna yönelik servisler (data access service), ihtiyaç duyulan uygulamanın gerçekleştirilmesini sağlayan servisler (Geo Information & Decision Support Service), belirli kullanıcı gruplarının konumsal portal üzerinde işlem gerçekleştirmesinin sağlanması için kimlik doğrulama ve yetkilendirme servisi (Authentication & Authorisation Services) olarak sınıflandırılabilir.

Şekil 35’de görülen GENESIS mimarisinde, Konumsal Bilgi & Karar Destek Servisler kısmında, Data Fusion, Geo Information ve Decision Support servisleri, WPS

tarafından gerçekleştirilmiştir. Bu çalışma kapsamında, bu servislerin gerçekleştirilmesi için WPS işlemleri geliştirilmiştir (Şekil 36).



Şekil 36. GENESIS projesi kapsamında kullanılan WPS işlemleri

GENESIS konumsal portal geliştirimi kapsamında, çeşitli prototip çalışmalar yapılmıştır. Londra şehrine yerleştirilmiş sensörler vasıtasıyla gelen verinin konumsal analiz servisleri ile işlenmesi ve analizi ile Londra şehrindeki hava kalitesinin tespiti ve harita arayüzü vasıtasıyla bu portal üzerinden sunulması uygulaması, gerçekleştirilmiştir. Constance gölüne yerleştirilen sensörler vasıtasıyla gelen verinin, konumsal analiz servisleri ile işlenmesi ve analizi ile bu göldeki su kalitesinin tespiti ve harita arayüzünde bu portal üzerinden sunulması uygulaması da gerçekleştirilmiştir (GENESIS, 2011).

2.6.3. AWARE ve GENESIS Konumsal Portallarının İrdelenmesi

AWARE konumsal portalı, sunucu tarafında ve kullanıcı tarafında çeşitli eksiklikleri bulunmaktadır. Öncelikle, sunucu tarafında geliştirilmiş olan WPS işlemleri WPS 0.4.0 standardı ile uyumludur. WPS 0.4.0 standardı güncellenerek, WPS 1.0.0 standardı yayınlanmıştır. Geliştirilen bu WPS işlemleri bu standard ile uyumlu değildir. Ayrıca,

WPS işlemleri, kompozit WPS işlemlerdir. WPS işlemlerinin kompozisyonu ile daha karmaşık modeller atomik düzeyde WPS işlemleri ile gerçekleştirilebilir. Böylelikle, WPS işlemlerinin yeniden kullanılabilirlik seviyesi artırılabilir. AWARE konumsal portalının kullanıcı tarafında ise, geliştirilen portal arayüzü, bu tez kapsamında ayrıntılı bir şekilde anlatılan JSR-286 ve WSRP standartları ile uyumlu değildir. Bu açıdan portal, büyük bir dezavantaja sahiptir.

GENESIS konumsal portalı, konumsal portal yazılımları içerisinde JSR-286 standardını sağlayan, az sayıda portal uygulamalarından birisidir. Yeniden kullanılabilir portletlerin geliştirilmiş olmasıyla, bu konuda avantaja sahiptir. Fakat, JSR-286 standardına uyumlu olarak geliştirilen bu portletler, WSRP olarak yayınlanmamıştır. GENESIS yaklaşımında, çeşitli tematik çevresel sorunlara çözüm olabilmesi öngörülen bu portal mimarisinin, portlet uygulamalarının WSRP olarak yayınlanmasıyla, bu uygulamalara uzaktan erişim ile kullanma ihtiyacı doğabilecek diğer portal uygulamaları açısından faydalı olacaktır.

2.6.4. Katalog Konumsal Portal ve Uygulama Konumsal Portalının Karşılaştırılması

Uygulama tabanlı konumsal portallara örnek bu iki uygulama, bu yaklaşımın önemini ortaya koymaktadır. KVA çalışmalarının hız kazanması ile konumsal veri hacminin hızla artması, konumsal veri çeşitliliğinin artması sebep olmuş ve ihtiyaç duyulan veriye ulaşmayı güçleştirmiştir. Katalog konumsal portal yapısı, tam anlamıyla birlikte işlerlik mekanizmalarının çalışmasını sağlayamamıştır. Bu yaklaşım, konumsal veriye ulaşma ve konumsal veriyle iş yapma bağlamında zaman ve kaynak kaybına yol açmakta ve konumsal veri hacminin ve çeşitliliğinin artması ile çoğu zaman konumsal veriye ulaşamamaktadır.

Bu tarzdaki portal uygulamaları yerine, tematik bir amaç doğrultusunda, belirli kullanıcı gruplarına hitap eden ve bu kullanıcı gruplarının ihtiyaçlarını karşılayacak konumsal analiz rutinlerinin portal ortamına entegrasyonunun sağlanması ile gerçekleştirilecek uygulama tabanlı portallar, birlikte işlerlik mekanizmalarını daha etkin bir şekilde gerçekleştirecektir.

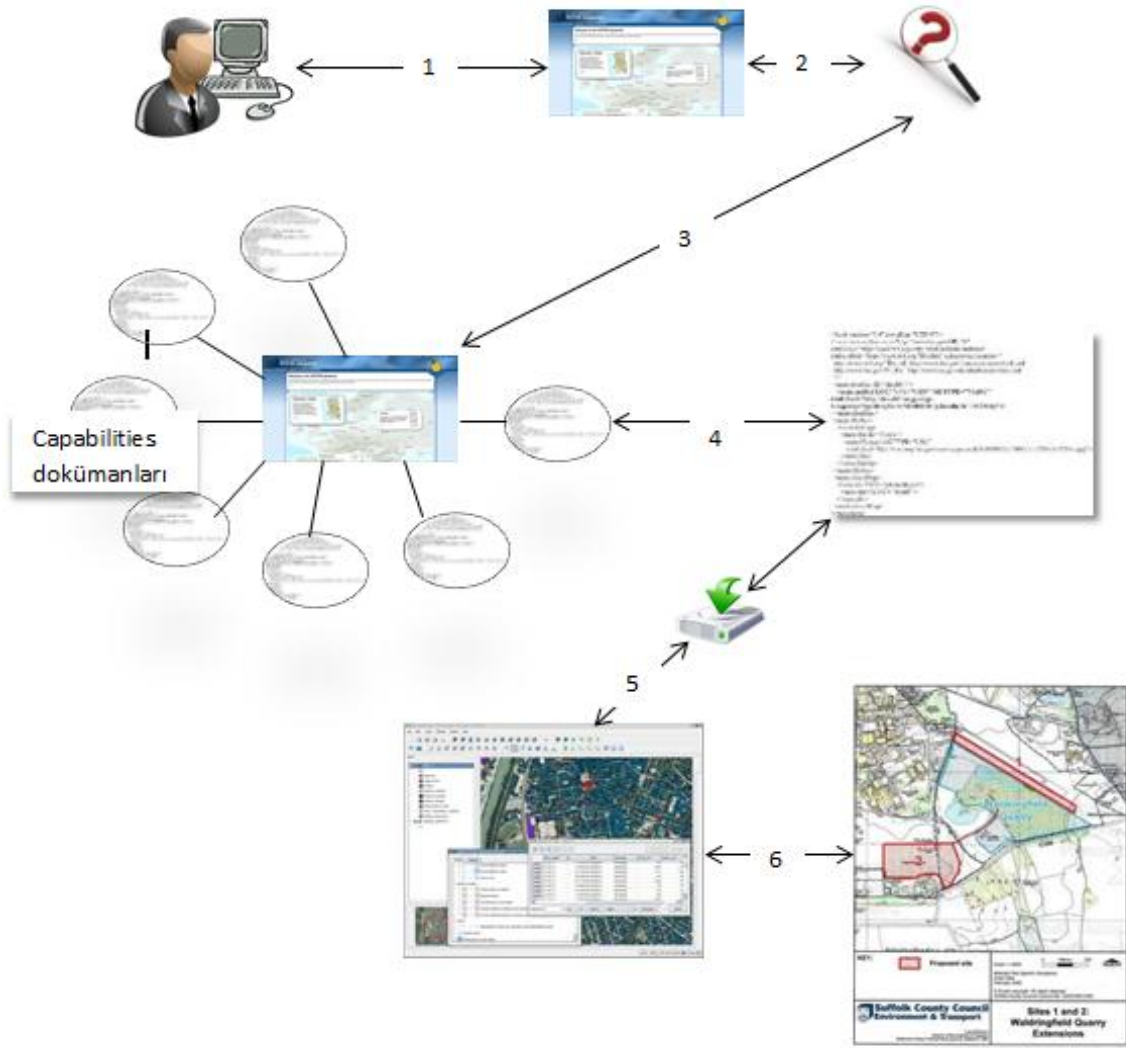
Bu çalışma kapsamında, portal teknolojisinin UKVA gerçekleştirimi bakımından sunacağı katkı incelenmiştir ve bu konuda yeni yaklaşımlar ortaya konulmuştur. Bu yaklaşımlardan bir tanesi, UKVA ortamında portalların, konumsal veriye ihtiyaç duyan kullanıcı ile konumsal veri sunucuları arasında bir aracı olmasının yanısıra kullanıcının ihtiyaç duyduğu konumsal veri üzerinde CBS analizlerini, portal ortamında gerçekleştirebildiği bir çerçeve(framework) ortaya koymaktır. İkinci yaklaşım ve bu çalışmada ayrıntılı bir şekilde nasıl gerçekleştirilebileceği gösterilen, portal ortamında portletler vasıtasıyla web servislerinin kompozisyonunun gerçekleştirilmesidir.

Katalog konumsal portal mimarisinde, bir kullanıcının uygulama geliştirme işlem adımları Şekil 37’de gösterilmiştir. Şekildeki işlem adımları aşağıdaki gibidir:

1. Kullanıcı, internetten konumsal portal uygulamasına girer. Konumsal portal, kullanıcı adı ve şifre ister.
2. Kullanıcı adı ve şifre girildikten sonra, kullanıcı, konumsal portalda, ihtiyaç duyduğu veri/verisetleri için metadata standartlarına göre arama yapar.
3. Arama sonucunda, çok sayıda sonuç portal sayfasından kullanıcıya sunulur.
4. Kullanıcı, çıkan bu sonuçların her birinin metaverisini inceler. İncelediği metaveriler içerisinde ihtiyaç duyduğu veri/veriseti olması olasılığı yüksek sonuçlar için birer birer Getcapabilities isteği gönderir. Capabilities dokümanlarının incelenmesinden sonra, uygun olan veri/veri setleri için sağlayıcının sunucusuna indirme isteği gönderilir.
5. Arama sonucunda bulunan veri uygulamanın gerçekleştirileceği ortama indirilir. Uygulama herhangi bir CBS yazılımı (Quantum GIS, gvSIG vb.) ile gerçekleştirilir.
6. Sonuç olarak, sonuç harita üretilir ve ilgili kullanıcılara sunulur.

Uygulama tabanlı konumsal portal mimarisinde, bir kullanıcının uygulama işlem adımları aşağıdaki gibidir.

1. Kullanıcı, internetten konumsal portal uygulama web sayfasına herhangi bir tarayıcı vasıtasıyla girer. Konumsal portal, kullanıcı adı ve şifre ister.
2. Kullanıcı adı ve şifre girildikten sonra, kullanıcı, konumsal portal ihtiyaç duyduğu analizi konumsal portal üzerinden gerçekleştirir.
3. Konumsal analizin sonucu ilgili kullanıcılara portal üzerinden sunulur.



Şekil 37. Katalog Konumsal Portal kullanılarak uygulama gerçekleştirmek isteyen kullanıcının işlem adımları

Arama sonucunda tüm sonuçlar için Capabilities sorgusunun yapılması ve Capabilities dokümanlarının incelenmesi uzun zaman almaktadır. Sonuç sayısının fazla olması ihtiyaç duyulan verinin bulunmasına imkân vermediği durumlarda ortaya çıkarmaktadır. Katalog konumsal portal ortamında uygulama geliştirecek kullanıcının, ihtiyaç duyduğu konumsal verilere ulaşma süresi göz önüne alındığında, hizmet kalitesinin de düşük olduğu görülmektedir.

Kullanıcı, ihtiyaç duyduğu veriye ulaşabilmek için, OGC mimarisinde yeterli düzeyde bilgi sahibi olması gerekmektedir. Örneğin; kullanıcının ihtiyaç duyduğu veri bir WFS sunucusu tarafından sunuluyorsa, kullanıcının WFS sunucusuna nasıl Capabilities

isteği gönderileceğini ve bu istek sonucunda sunucu tarafından döndürülen Capabilities dokümanını, nasıl incelenmesi gerektiğini bilmesi gerekmektedir. Dolayısıyla, bu mimari kullanıcılara ek külfetler getirmektedir.

2.6.5. Uygulama Tabanlı Prototip Konumsal Portal Geliştirimi

Uygulama tabanlı konumsal portal ortamında, belirli sorunların çözümüne yönelik konumsal portalların oluşturulması, kullanıcıların ihtiyaç duyduğu veriye kısa sürede ulaşmasını sağlayacak, ihtiyaç duyduğu konumsal analizi portal ortamında gerçekleştirerek, sadece verinin kullanıcılara sunulması değil, uygulamanın gerçekleştirilmesiyle, hizmet kalitesi artırılacak ve UKVA ortamında oluşturulması hedeflenen “birlikte işlerlik” altyapılarına destek sağlayacaktır.

Bu çalışma kapsamında, uygulama tabanlı konumsal portala örnek teşkil edecek portal geliştirimi gerçekleştirilmiştir. Bu portalda, kullanıcılar günümüzde büyük bir sorun olan katı atık depolama tesisi yapılamayacak alanların bulunması analizini, gerçekleştirebilmektedir. Şekil 38’ de geliştirilen portal mimarisi ve bileşenleri görülmektedir. Bu portal mimarisi beş bileşene ayırmak mümkündür. Bu bileşenler; Konumsal portal sunucusu, Servis kompozisyonu, WPS, Konumsal Veri Sunumu ve Konumsal Veri olarak adlandırılmıştır.

Konumsal Portal Sunucusu: portal mimarisinin bu bileşenin de, geliştirilen portletler için gerekli olan koşum-zamanı ortamı (runtime environment) sağlanmaktadır. Portal kullanıcılarının, konumsal veri üzerinde arama,izleme, indirme ve konumsal analizleri gerçekleştireceği arayüz, portal sunucusu tarafından sağlanmaktadır. Bu çalışmada geliştirilen bufferMEY, intersectMEY, unSuitZoneMEY ve bufferIntRoadMEY processlerine arayüz oluşturan dört adet portlet tasarlanmıştır. Bunlar buffer, intersect, unsuitzone ve bufferIntRoad portletleridir. Geliştirilen portletler, Yıldırım ve Cömert (2011)’ de gösterildiği üzere, yeniden kullanılabilir uygulamalardır. Çalışmada, portal sunucusu olarak açık kaynak kodlu yazılım olan Liferay Portal sunucusu kullanılmıştır.

Servis Kompozisyonu: Bu çalışmada, servis kompozisyonu için üç yaklaşım kullanılmıştır. Bunlar; “Portlet Eventing” ile servis kompozisyonu, BPEL motoru ile servis

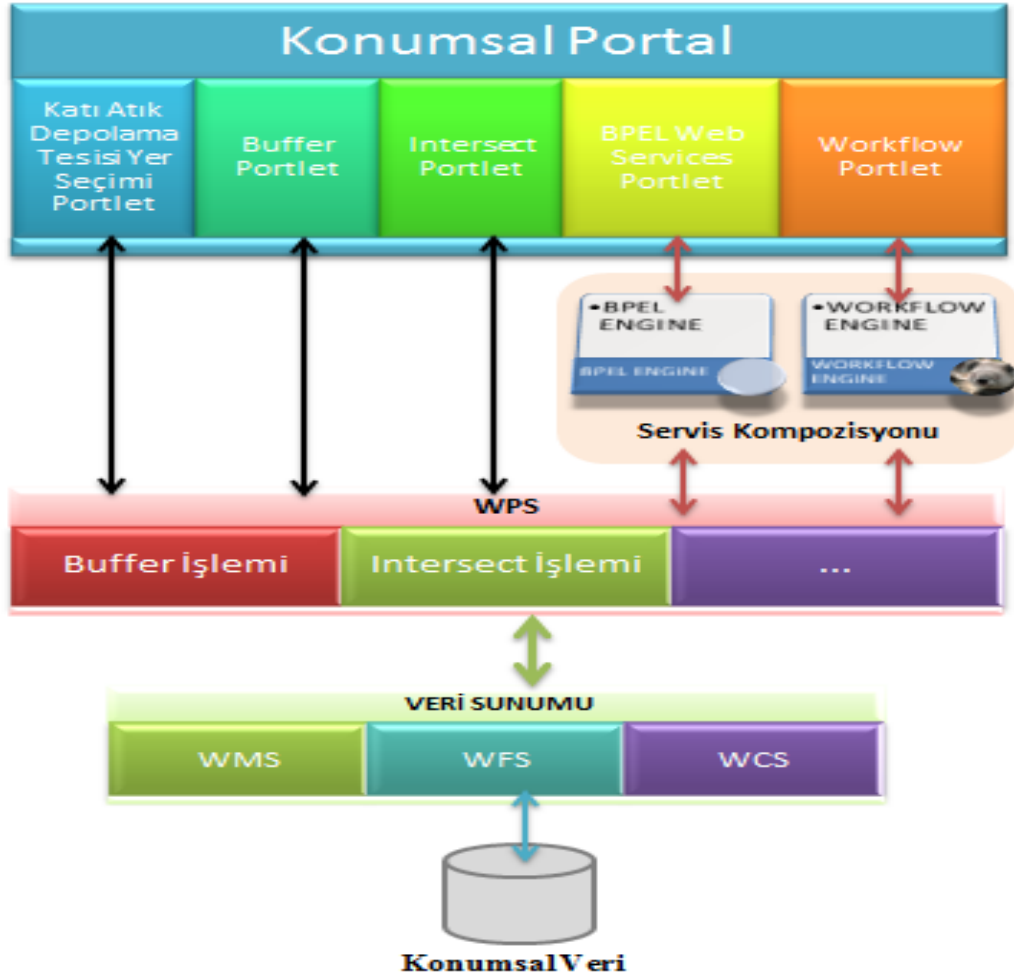
kompozisyonu ve bir iş akış motoru ile servis kompozisyonudur. “Portlet Eventing” ve Web Servisi Kompozisyonu bölümünde anlatıldığı gibi “Portlet Eventing” mekanizması kullanılarak web servisi kompozisyonu sağlanmıştır. Bu uygulamada, kullanıcı buffer portleti vasıtasıyla WPS sunucusuna buffer isteği gönderir. Bu istek sonucunda, katı atık yer seçimi analizinde buffer uygulanacak verileri, WFS linki olarak portlet eventing yoluyla, Intersect portletine gönderilir. Intersect portleti, buffer uygulanmış tüm katmanları kullanarak intersect analizi gerçekleştirir. unSuitZoneMEY, portleti ise katı atık yer seçimi için uygun olmayan bölgenin sınır verisini, buffer ve intersect portletinin çıktı verisini kullanarak üretmektedir. Bu servis kompozisyonu işlemi portal ortamında portletler vasıtasıyla gerçekleştirilmiştir. İkinci yaklaşım olan, BPEL motoru ile servis kompozisyonu yaklaşımı bu konumsal portal uygulamasında gerçekleştirilmiştir. WPS işlemlerinin WSDL tanımlarının kullanarak kompozisyon işlemi gerçekleştirilmiştir. BPEL Web Services portleti bu uygulamaya arayüz olarak geliştirilmiştir. Üçüncü yaklaşım olan, bir iş akış motoru olan Taverna Workbench 2.0 workflow motoru ile WPS işlemlerinin kompozisyonu gerçekleştirilmiştir. Bu uygulama için Workflow portleti geliştirilmiştir.

WPS: Portal mimarisinin bu bileşenin de, konumsal veri üzerinde gerçekleştirilecek konumsal analizler için geliştirilen WPS işlemlerinin bir WPS sunucusu tarafından sunulduğu bileşendir. Bu bileşen için geliştirilen WPS işlemleri, 2.4. WPS ile Web Tabanlı Konumsal Analiz başlığı altında ayrıntılı bir şekilde açıklanmıştır. Bu bileşen için açık kaynak kodlu yazılım olan Degree WPS sunucusu kullanılmıştır.

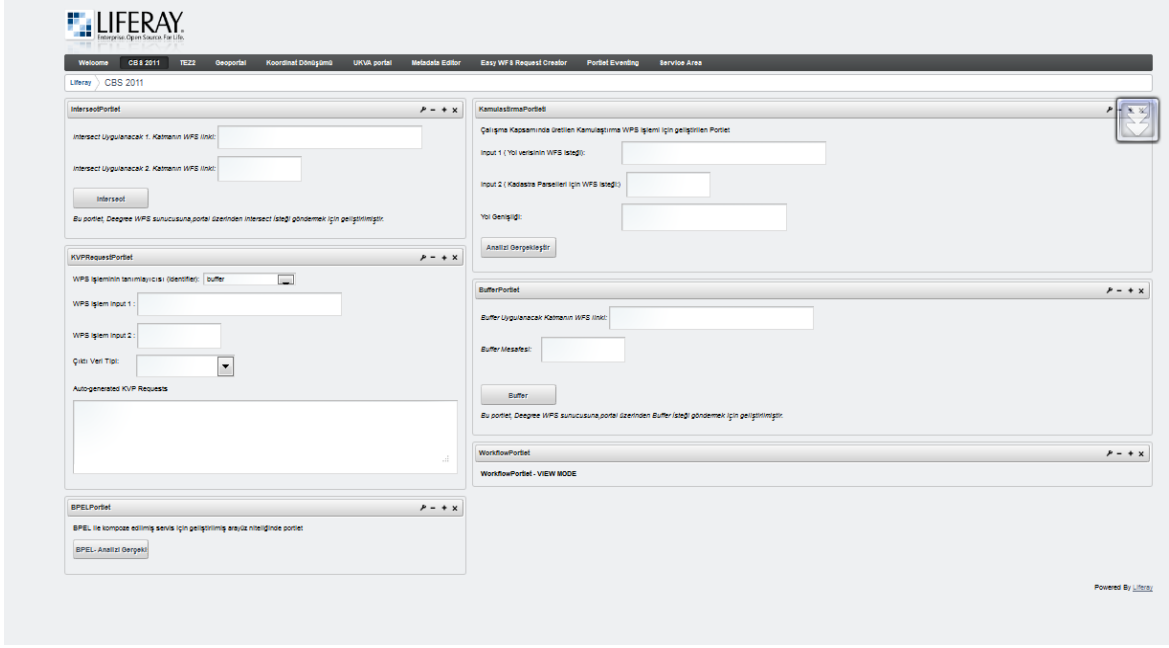
Konumsal Veri Sunumu: Portal ortamında gerçekleştirilecek konumsal analizler için kullanılacak verilerin, sunucular tarafından sağlanması öngörülmüştür. Dolayısıyla, bu portal ortamında gerçekleştirilecek uygulamalarda kullanılan veriler, konumsal veri sunucuları tarafından sunulmaları gerekmektedir. OGC tarafından konumsal verilerin web üzerinden sunulmasına yönelik web servisleri bulunmaktadır. Bunlar; Web Feature Servisi (WFS), Web Map Servisi (WMS) ve Web Coverage Servisi (WCS) dir. Bu portal bileşeninin gerçekleştirilmesinde, açık kaynak kodlu yazılım olan Geoserver 2.1.0 kullanılmıştır.

Konumsal Veri: Bu çalışma kapsamında geliştirilen portal mimarisinin bu bileşeni en temel bileşendir. Uygulama için gerekli tüm verileri kapsamaktadır. Uygulamanın gerek

duyduğu verilerin, ilgili kurumlardan Web servisleri ile “anlık” olarak sağlanması öngörülmüştür. Bunun mümkün olamayacağı kurumlar için ise bu yapı uyarlanmıştır.



Şekil 38. Çalışma kapsamında geliştirilen konumsal portal mimarisi



Şekil 39. Portaldan Ekran görüntüsü

2.7. Konumsal Portal Geliştiriminde Açık ve Özgür Kaynak Kodlu Yaklaşım

UKVA ortamında hızlı portal geliştirimini destekleyecek bileşenlerden birisi de, açık ve özgür kaynak kodlu yaklaşımdır. Portal uygulamalarının ticari yazılımlarla geliştirilmesi, ticari yazılımlara bağıllığı artıracaktır. İlerleyen kısımda, ticari yazılımlarla portal gerçekleştirimi ayrıntılı bir şekilde irdelenecektir. Şüphesiz, açık ve özgür kaynak kodlu yaklaşım, portal geliştirim maliyetini azaltacaktır. Bu yaklaşımla birlikte, ticari yazılımlara bağıllık ortadan kaldırılacak, portal geliştirim maliyeti düşecek ve hızlı portal geliştirimi desteklenecektir.

Bu çalışma kapsamında dile getirilen konumsal portal üretiminde açık ve özgür kaynak kodlu yaklaşımın anlaşılabilirliği için açık ve özgür yazılım kavramlarının anlaşılmasında fayda vardır. Çoğu zaman bu kavramlar birbirine karıştırılmaktadır.

Özgür yazılımın tanımı, Richard Stallman tarafından yapılmış ve Özgür Yazılım Vakfı (Free Software Foundation) tarafından 1986 yılında yayınlanmıştır (URL 23). Bu tanımda, özgür kavramı felsefi olarak, bedava içecek olarak değil, konuşma özgürlüğü kavramındaki özgürlük olarak düşünülmelidir (URL 24). Özgür yazılımın tanımındaki

“Özgür yazılım bir özgürlük meselesidir, fiyat değil” cümlesi, özgür yazılım felsefesini iyi bir şekilde tanımlamaktadır.

Özgür yazılım kavramı, kullanıcıların, yazılımı çalıştırma, kopyalama, dağıtma, üzerinde çalışma, değiştirme ve geliştirme özgürlükleriyle ilgili bir kavramdır. Daha açık ifade edilirse, "özgür yazılım" kavramında, yazılımlar aşağıdaki özgürlüklere sahip olmalı gerekmektedir:

- Herhangi bir amaç için yazılımı çalıştırma özgürlüğü,
- Her ne istiyorsanız onu yaptırmak için programın nasıl çalıştığını öğrenmek ve onu değiştirme özgürlüğü:Yazılımın kaynak koduna ulaşmak, bu iş için önkoşuldur,
- Kopyaları dağıtma özgürlüğü: Böylece komşunuza yardım edebilirsiniz,
- Tüm toplumun yarar sağlayabileceği şekilde programı geliştirme ve geliştirdiklerinizi (ve genel olarak değiştirilmiş sürümlerini) yayınlama özgürlüğü
Kaynak koduna erişmek, bunun için bir önkoşuldur.

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 (URL 29).

Açık kaynak ile özgür kaynak kavramları birbirine yakın olsalarda tamamen aynı değildir. Literatürde, açık kaynak ve özgür kaynak arasındaki farkların irdelendiği, (URL 25), (URL 26), (URL 27) gibi çalışmalar mevcuttur. Özgür kaynağı tanımlayan Richard Stallman, açık kaynak olmayan yazılımları sosyal sorun olarak tanımlamakta ve açık 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 (CIOOC, 2009). Ayrıca, hükümetimiz tarafından 2011 yılında yayınlanan duyuruda kamu kurum ve kuruluşların, açık ve özgür kaynak kodlu yazılımları kullanması tavsiye edilmiştir.

Açık ve özgür kaynak kodlu yaklaşım, portal ürünleri için hayati önem taşımaktadır. Konumsal portallar, konumsal veri ile iş yapan kesimler arasındaki birlikte işlerliğine yönelik geliştirilen teknolojilerdir. Tüm dünyada olduğu gibi, ülkemiz için hayati bir önem taşımaktadır. Konumsal portal geliştiriminin, tez kapsamında anlatıldığı gibi yeniden kullanılabilir uygulamalarla, standartlara uyumlu ve farklı portallar arasında birlikte işlerliğin sağlanabildiği bir ortamla hızlı, kaliteli ve etkin bir şekilde yapılması gerekmektedir. Bu da ancak bu yaklaşım ile tez kapsamında ortaya koyulan çerçevenin tamamlanmasıyla oluşturulabilecektir. Nitekim, bu yaklaşımın önemini ortaya koyan bir değişim, ESRI firması tarafından gerçekleştirilmiştir. ESRI firmasının portal yazılımı olan GeoPortal Server yazılımı, ilk etapta ücretli bir yazılım iken, firma tarafından açık ve özgür kaynak kodlu olarak yayınlanmaya başlanmıştır. Bu da yazılımın herkes tarafından geliştirilmesini sağlamıştır. ESRI firmasının portal ürününü açık kaynak kodlu olarak yayınlanmasının sebeplerinin irdelenmesi, açık kaynak kodlu yaklaşımın, sahip olduğu avantajları da ortaya çıkaracaktır. Bu sebeplerden bir tanesi, bu alandaki açık kaynak kodlu yazılımlarının çok sayıda ve güçlü olması, ESRI firmasının böyle bir karar almasında etkili olmuştur. Açık kaynak kodlu yazılımların, hızla gelişmesi bu alanda hatırı sayılır bir piyasa yaratmıştır. Ticari firmalar, bu piyasadan uzak kalmak istememeleri sonucunda, ticari yazılımlarda, günümüzde açık kaynak kodlu projelere destekler verdikleri görülmektedir. ESRI firması bu konudaki yaklaşımını, kapalı kod ve açık kaynak kodlu yazılımların entegrasyonunu sağlayan “hybrid” modeli olarak adlandırmaktadır (URL 28). Bu yazılım ilk etapta ArcGIS Server yazılımı için eklenti (extension) olarak geliştirilmiştir. Bu eklenti daha sonra, Apache 2.0 lisansı altında açık kaynak olarak yayınlanmıştır. Bu yazılım her ne kadar, açık kaynak kodlu sunucularda çalışabildiği iddaa edilse de, bu eklentinin açık kaynak kodlu yazılımlarla entegrasyonunda sıkıntı yaşanmaktadır. Ticari yazılım olan ArcGIS server ortamında, bu portal yazılımının kullanımında herhangi bir sorun yaşanmamaktadır. Bu ürün kullanılarak yapılan portal gerçekleştiriminin büyük çoğunluğu ArcGIS Server ortamında gerçekleştirilmiştir. Her ne kadar ESRI firmasının portal ürünü, ücretsiz ve açık kaynak kodlu olsa da, bu bağlamda bir pazarlama politikası olarak

değerlendirilebilir. Burada, ESRI firmasının yapmaya çalıştığı, açık kaynak kodlu yazılımlarının gücünü kullanarak, yazılımın pazarlanması olarak değerlendirilebilir.

Bir yazılımın açık kaynak kodlu olarak yayınlanması ve geliştiriciler tarafından geliştirilmesi, ticari yazılım grubu tarafından geliştirilmesine göre bir üstünlük sağlar mı? sorusunun cevabı, açık ve özgür kaynak kodu yaklaşımının avantajını ortaya koymaktadır.

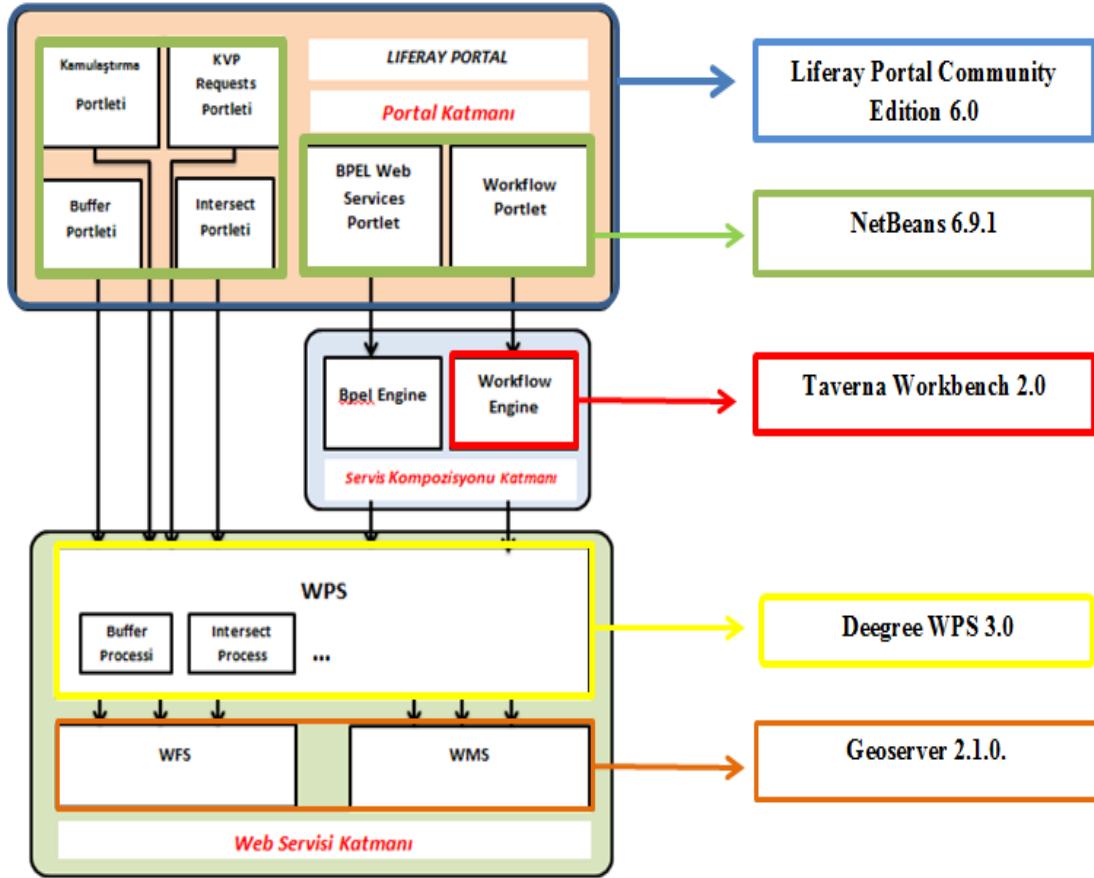
Ticari yazılım grupları tarafından geliştirilen yazılımlar, belirli sayıda geliştirici tarafından geliştirilen yazılımlardır. Her ne kadar ticari yazılım grubunun gücüne göre yazılımcı sayısı artsa da, açık kaynak kodlu yazılımlarda yazılımcı sayısında, belli bir sınırlama söz konusu değildir. Konu üzerinde bilgi sahibi olan geliştiriciler, her an geliştirmelerini yaparak yazılımda geliştirmeler yapabilmektedir. Kapalı kod yazılımlarda geliştirmeler, yazılım grubunun yaptığı geliştirmelerle sınırlıdır. Oysaki açık ve özgür kaynak kodlu yazılımların kullanıcıları, hem yazılımın kullanıcıları, hemde tespit ettikleri eksiklikler için yazılımda geliştirmeler yaparak geliştirici olabilmektedir.

Kapalı kod yazılımların dezavantajlarından birisi de, güvenilirlik konusudur. Kamu kurum ve kuruluşlarında kullanılan yazılımların arka planda nasıl işlemler yapıldığının bilinmemesi, yazılımların güvenilirliğini azaltmaktadır. Kullanıcılar, ihtiyaçları doğrultusunda istedikleri gibi kapalı kod yazılımları üzerinde düzenleme gerçekleştirilememektedir. Açık ve özgür kaynak kodlu yazılımlarda ise kullanıcı ihtiyacı doğrultusunda istediği değişikliği yapabilmektedir. Bu şekilde, yazılımların geliştirilmesi aşamasının daha şeffaf olması sağlanmıştır. Açık kod yazılımlarda, kodun bir çok geliştirici tarafından incelenebilmesi ve eksiklerin herhangi bir geliştirici tarafından geliştirilebilmesi, yazılımın kalitesini de artırmaktadır.

Ülkemiz için hayati önem arz eden UKVANın gerçekleştirilmesi aşamasında kullanılacak yazılımların, ticari yazılım grupları tarafından üretilen yazılımlar olması yerine, geliştirme aşamaları tamamen şeffaf, bir çok yazılımcı tarafından incelenebilen, daha düşük maliyetlerle üretilen açık ve özgür kaynak kodlu yazılımlar olması, sürecin daha iyi yönetilebilmesini sağlayacaktır. İhtiyaca yönelik noktasal çözümlerle, sürecin hızlandırılması sağlanabilecektir.

Günümüzde, UKVA konumsal portallarına yönelik sorunu 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 portalın 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 40. Portal Mimarisi ve kullanılan açık kaynak kodlu yazılımlar

Liferay Portal Community Edition 6.0 yazılımı bu çalışmada portal sunucusu olarak kullanılmıştır. NetBeans IDE 6.9.1 yazılımı, portlet geliştirme ortamı olarak kullanılmıştır. Taverna Workbench 2.0 yazılımı iş akış motoru olarak kullanılmıştır. Deegree WPS 3.0 ise wps işlemlerinin geliştirilmesi ve web üzerinden analizlerinin gerçekleştirilmesi için kullanılmıştır. Geoserver 2.1.0 sunucusu ise WFS ve WMS servislerini kullanarak web üzerinden veri sunulması için kullanılmıştır. OpenLayers ise Harita Görüntüleyici portletinde açık kaynak kodlu web tabanlı harita arayüzü olarak kullanılmıştır.

Bu tez kapsamında yapılan çalışmanın bu kısmında, tamamı açık ve özgür kaynak kodlu yazılımlar ile konumsal portal geliştiriminin yapılabileceğinin ispat edilmesi, önemli bir sonuçtur. Bu yaklaşım, konumsal portal geliştiriminin, “hızlı”, “kaliteli” ve “az maliyetli” yapılabilmesine olanak sağlaması açısından, önemli bir yaklaşımdır.

3. BULGULAR VE İRDELEMELER

Bu tez çalışmasının çıkış noktası, halen ülkemizde çözüm bulunamamış konumsal portal geliştirimine yönelik yeni yaklaşımların ve teknik mimarinin ortaya koyulmasıdır. Bu çalışmada, birlikte işlerliğin sağlanmasına yönelik gerçekleştirilecek konumsal portalın, hangi teknolojiler, standartlar ve yaklaşımlar ile birlikte gerçekleştirileceği belirlenmeye çalışılmıştır. Yapılan çalışmalar sonucunda, konumsal portallar için üç bileşen ortaya koyulmuş ve bu doğrultuda prototip konumsal portal geliştirimleri yapılmıştır. Bunlardan ilki; UKVA ortamında hızlı portal geliştirmedir. İkincisi; uygulama odaklı konumsal portal geliştirmesi ile kullanıcıların konumsal portal ortamında konumsal analizleri gerçekleştirilmesinin sağlanmasıdır. Üçüncü bileşen ise; konumsal portal geliştirmede açık ve özgür kaynak kodlu yaklaşımdır.

Dünya genelinde yapılan UKVA konumsal portal geliştirmeleri incelendiğinde, portal uygulamalarının büyük bir çoğunluğunun yeniden kullanılabilir nitelikte uygulamalar olmadığı görülmektedir. Bu konumsal portalların bir kısmının da, yeterli düzeyde yeniden kullanılabilir olmayan uygulamalar, gerçekleştirildiği görülmektedir. UKVA ortamında, hızlı portal geliştirmesi son derece önemli bir sorun teşkil etmektedir. Özellikle, Türkiye gibi dünyadaki UKVA çalışmalarından geri kalan ülkeler, konumsal portallarını kısa sürede gerçekleştirmeye ihtiyaç duymaktadırlar. Dolayısıyla hızlı portal geliştirmesinin sağlanması için geliştirilecek uygulamaların yeniden kullanılabilir özellikte ve kolayca portal çalışma ortamına entegre edilebilir olması gerekmektedir (Yıldırım ve Cömert, 2011).

Konumsal olmayan portal uygulamalarında, yeniden kullanılabilirlik sorunu büyük önem taşırken ve bu sorunun çözümüne yönelik çalışmalar mevcut iken, konumsal portal uygulamaları alanında, bu sorun göz ardı edilmiştir. Konumsal portal uygulamalarında, portal ortamında yayınlanan ve kullanılan veri/veriseti veya servisler için bir çok standart geliştirmesinin yapılmasına rağmen, UKVA ortamının önemli bileşeni olan konumsal portalların geliştirilmesine yönelik herhangi bir standart kullanımı söz konusu değildir. Konumsal portallardaki yeniden kullanılabilirlik sorununa çözüm için, konumsal portalların belli standartlara göre geliştirilmesi gerekmektedir. Yeniden kullanılabilirliğin

sağlanması için konumsal portalların geliştirilmesinde kullanılması gereken standartlar, JSR-286 ve WSRP standartlarıdır.

Konumsal portal geliştiriminde bu iki standardın kullanılmasıyla, konumsal portal uygulamalarında söz konusu olan yeniden kullanılabilirlik sorunu çözülebilecektir. Konumsal portallardaki bu sorunun çözülmesiyle, hızlı portal geliştirimi sağlanacaktır. Bu çalışmada, ayrıntılı bir şekilde anlatılan hızlı portal geliştirimi yaklaşımı, ülkemiz gibi UKVA çalışmalarında geri kalmış ülkelerde, UKVA gerçekleştirimi için şu an ki en geçerli çözümü sunan konumsal portalların bir an önce hayata geçirilmesini sağlayacak ve ülke genelinde çeşitli ölçeklerde gerçekleştirilecek olan konumsal portalların hızlı geliştirimine destek sağlayacaktır.

Bu yaklaşım kapsamında, “yeniden kullanılabilir konumsal 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 çalışma kapsamında, ülkemizdeki konumsal portal çalışmaları da incelenmiştir. T.C. Orman ve Su İşleri Bakanlığının ve T.C. Çevre ve Şehircilik Bakanlığının bu konuda çalışmaları bulunmaktadır. T.C. Orman ve Su İşleri Bakanlığı tarafından yapılan portal uygulamasının, şu an aktif olmadığı görülmüştür. GEODATA VD4 adlı yazılımının, her ne kadar Coğrafi Veri Portalı olarak adlandırılırsa da, harita görüntüleme arayüzü olmaktan, ileri gidememiş bir uygulama olduğu görülmektedir. T.C. Çevre ve Şehircilik Bakanlığının Mekansal Veri Bilgi Sistemi olarak adlandırdığı çalışmada, harita görüntüleme arayüzü olmaktan ileri gidememiş bir uygulamadır. Görüldüğü üzere, ülkemizde konumsal portal olarak tanımlanabilecek bir uygulama söz konusu değildir.

Dünya genelindeki UKVA çalışmaları incelendiğinde, geliştirilen konumsal portalların katalog konumsal portalları olduğu görülmektedir. Bu mimaride, bir kullanıcının uygulama geliştirmesi için ihtiyaç duyduğu veriye ulaşabilmesi süresi ve yöntemi göz önüne alındığında, servis kalitesinin düşük ve kabul edilebilir düzeyde olmadığı görülmektedir. Kullanıcı, konumsal portal ortamında ihtiyaç duyduğu veri için arama yapmakta ve bulduğunda çalıştığı ortamına indirmekte ve herhangi bir masaüstü

CBS yazılımı ile uygulamayı gerçekleştirmektedir. Çoğu zaman, konumsal veri için arama yapma süresi saatleri, bazen günleri almakta ya da mümkün olmamaktadır. Ayrıca bu mimari, kullanıcıya ek külfetler getirmektedir. Bu mimaride, konumsal veriye ulaşacak kullanıcının, ulaştığı sonuçlar için Capabilities isteğini nasıl göndereceğini, konumsal veri sunucusundan gelen yanıt Capabilities dokümanının, nasıl inceleneceği ve ihtiyaç duyduğu veriyi indirmek için sunucuya nasıl istek göndereceğini, bilmesi gerekmektedir. Bu yaklaşım, hizmet kalitesi bakımından yeterli değildir.

Katalog konumsal portal mimarisi, UKVA ortamında günümüzün ihtiyaçlarını karşılayamamaktadır. Portal kullanıcıları, ihtiyaç duyduğu konumsal veriye hızla ulaşabildiği, konumsal veri üzerinde analizler gerçekleştirebildiği, portal ortamında ihtiyaç duymaktadır.

Bu tez kapsamında, bu sorunun çözümüne yönelik çalışmalar yapılmıştır. Konumsal portal ortamında, kullanıcıların konumsal veri üzerinde konumsal analizler gerçekleştirebilmesi sağlanmıştır. Bu doğrultuda, prototip uygulama odaklı konumsal portal geliştirimi gerçekleştirilmiştir. Prototip konumsal portal geliştirimiyle, uygulama odaklı konumsal portalın gerçekleştirilmesi için gereksinimler, hangi standartların ve teknolojilerin kullanılması gerektiği belirlenmeye çalışılmıştır.

Bu çalışmada, uygulamanın gerek duyduğu verileri web servisleri ile anlık olarak alması öngörülmüştür. Konumsal verinin sunumuna yönelik, OGC tarafından tanımlanan WMS, WFS ve WCS standartlarının yayınlanmasıyla, konumsal verilerin sunulmasına yönelik çok sayıda sunucu geliştirilmiştir. Bu çalışmada kullanılacak konumsal verilerin, ilgili kurumlar tarafından, WFS, WMS ve WCS servisleri ile sunulması öngörülmüş ve prototip uygulamalar bu öngörüye göre geliştirilmiştir.

Bu çalışmada, konumsal portal ortamında konumsal analizlerin gerçekleştirilmesi için, web tabanlı konumsal analiz imkanı sunan yazılımlar incelenmiştir. Bu çalışmada, iki yazılım irdelenmiştir. Bunlar; ArcGIS server ve NetGIS server yazılımlarıdır. Bu iki yazılımda, web üzerinden konumsal analiz gerçekleştirimi için yeterli düzeyde yazılımlardır. Fakat, bu yazılımlar kullanılarak geliştirilen uygulamalarda, başka yazılımlar ile entegrasyon sorunu ortaya çıkmaktadır. Örneğin; web tabanlı konumsal analizin

gerçekleştirildiği bir örnek uygulamanın bir kısmı NetGIS server, bir kısmı ArcGIS server, bir kısmı açık ve özgür kaynak kodlu yazılımlar ile yapılmak istenildiğinde, yazılımlar arasında entegrasyon gerçekleştirilememektedir. Dolayısıyla, yazılım bağımlılığını ortadan kaldırmak için konumsal portal ortamında, web üzerinden konumsal analiz gerçekleştirilmesi için bu ticari ve kapalı kod yazılımlar değil, OGC tarafından tanımlanan WPS standardı ve bu standartla uyumlu WPS sunucuları kullanılmıştır.

WPS standardı, web üzerinden konumsal analiz gerçekleştirimine yönelik OGC tarafından geliştirilmiş bir standarttır. Konumsal analiz, WPS işlemleri vasıtasıyla gerçekleştirilmektedir. Örneğin; kullanıcı, buffer analizini gerçekleştirmek istiyorsa, bir WPS sunucusunda, konumsal veriye buffer uygulayan bir WPS işlemi kullanacaktır. Bu şekilde, ihtiyaç duyulan analizler için WPS işlemleri geliştirilerek, web üzerinden konumsal analizler gerçekleştirilebilecektir. WPS ile konumsal portal entegrasyonu sağlanarak, kullanıcılara, konumsal portal ortamında konumsal analiz gerçekleştirebildiği bir ortam sağlanabilecektir.

Konumsal portal ortamında, karmaşık bir konumsal analizin gerçekleştirilebilmesi için SYM de olduğu gibi WPS işlemleri ile WSK gerçekleştirimine ihtiyaç duyulmaktadır. Bir WPS işlemi tarafından gerçekleştirilemediği durumda, WPS işlemlerinin kompozisyonu ile, portal ortamında, daha karmaşık konumsal analizlerin gerçekleştirilmesi sağlanacaktır.

Konumsal portal ortamında WSK gerçekleştirimi için, bu çalışmada dört yaklaşım ortaya koyulmuştur. Bu yaklaşımlar; “Portlet Eventing” ile WSK, KVP isteklerler az sayıda servisin kompozisyonu, yeni bir WPS işlemi tasarlama ile WSK, bir iş akış motoru ile WSK’dır. Bu dört yaklaşım için tez kapsamında uygulamalar gerçekleştirilmiştir. Literatürde, WPS kullanılarak yapılan uygulamalarda, yeni bir WPS işlemi tasarlama ile WSK, kompozit WPS olarak da adlandırılan yaklaşım çoğunlukla kullanılmıştır. “Portlet Eventing” ile WSK, konumsal portal alanında bu çalışmada ortaya koyulan özgün bir yaklaşımdır. Bu çalışmada WPS ile konumsal portal entegrasyonu sağlanmış ve kullanıcılara portal ortamında konumsal analiz imkanı verilmiştir.

“Portlet Eventing” ile WSK, WPS işlemlerinin kompozisyonu için portletlerin kullanıldığı bir yöntemdir. Bu tez kapsamında, bu yaklaşım için örnek çalışmalar yapılmıştır. Yeniden kullanılabilir portletler ile WSK yaklaşımı, konumsal portal alanında, yenilikçi bir yaklaşımdır. Bu uygulama ile portletler, konumsal portallarda sadece kullanıcı arayüzü olarak kullanılmamış, bu yaklaşıma ek olarak, portal ortamında konumsal analizin gerçekleştirilmesi için WPS işlemlerinin kompozisyonunda kullanılmıştır.

Bu çalışmada, ayrıntılı bir şekilde WPS standardı incelenmiş ve çeşitli sorunlar tespit edilmiştir. Bu sorunlar, standarttaki kavramsal sorunlardır. 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 kalınmıştır. 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 kavramı, 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? Dolayısıyla,

“basit” ifadesi yerine “az sayıda” ifadesinin kullanılması daha uygundur. Çünkü, KVP kodlanmış isteklerde sunucuya gönderilebilecek isteklerde 2048 karakter sınırlaması mevcuttur. KVP kodlanmış isteklerle, ancak az sayıda istek bu karakter sınırlamasından dolayı servis kompozisyonunda kullanılabilir.

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/identifier[/service.soap]?WSDL*” şeklinde tüm işlemler için ayrı ayrı sunmaları da tavsiye edilmiştir.

Bu çalışma kapsamında, 52 North WPS, Deegree WPS, Geoserver WPS, pyWPS sunucuları, WSDL dokümanını yayınlayan arayüzü sağlaması bakımından incelenmiştir. Geoserver WPS yazılımı haricindeki tüm WPS sunucuları standartta dile getirilen WSDL dokümanın sunulduğu arayüzü desteklemektedirler. Fakat sunucular WSDL dokümanını otomatik olarak değil, kullanıcıların kendilerinin oluşturmasıyla, sunabilmektedir. Bu yaklaşım, WPS sunucuları için bir eksiklik olarak göze çarpmaktadır.

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 üretilmesi sağlanmalıdır. 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 (*identifier*) 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 kaydolması yerine, işlemler ayrı ayrı kaydolabilecektir. Böylelikle, kullanıcılara servis kataloglarında, işlemlere göre arama yapma imkânı verilecektir. WPS işlemlerinin ayrı ayrı WSDL dokümanlarını ile servis kataloglarına kaydedilememesi, WPS işlemlerinin bir iş akışı ile WSKsı uygulamasına ek külfet getirmekte, geliştirici ihtiyaç duyduğu WPS işlemi için, tüm WPS için oluşturulan WSDL dokümanını incelemektedir.

Üçüncü bileşen olan konumsal portal geliştiriminin açık ve özgür kaynak kodlu yaklaşımı, bu çalışma kapsamında UKVA ortamında konumsal portal geliştirilmesine yönelik ortaya koyulan çerçeveyi tamamlayacak bir yaklaşımdır. Açık ve özgür kaynak kodlu yaklaşım bu çerçevede, iki aşamada kullanılmıştır. Birinci aşaması, konumsal portal geliştirimin açık ve özgür kaynak kodlu yazılımlar ile yapılmasıdır. Bu çalışmada, bir konumsal portal geliştiriminin, açık ve özgür kaynak kodlu yazılımlar ile gerçekleştirilebileceği ispat edilmiştir. Bu yaklaşımın ikinci olarak kullanıldığı yer ise konumsal portallar için geliştirilen uygulamaların, bir portlet kataloğu veya UDDI kataloğundan açık ve özgür kaynak kodlu yazılım olarak yayınlanmasıyla, hızlı portal geliştiriminin sağlanması aşamasıdır.

Konumsal portalların, açık ve özgür kaynak kodlu yaklaşımlarla yapılması, hiç şüphe yok ki, konumsal portal geliştirim maliyetini düşürecektir. Konumsal portal geliştiriminin, açık ve özgür kaynak kodlu yazılımlar ile geliştirilmesinde, kullanılacak portal bileşenleri, standartlar ve mimari belirlenmeye çalışılmıştır. Bu doğrultuda, prototip konumsal portal geliştirmeleri yapılmıştır. Bu tez kapsamında, geliştirilen tüm konumsal portal prototipleri, açık ve özgür kaynak kodlu yazılımlar ile yapılarak, bunun mümkün olabileceği ispatlanmıştır.

UKVA ortamında konumsal portlet kataloğunun oluşturulması ile UKVA ortamında kurumların geliştirdiği, konumsal portal uygulamalarındaki konumsal portletler, bu katalogdan yayınlanabilecektir. Konumsal portal geliştiricileri, bu katalogdan yayınlanan konumsal portlet kataloglarını kullanarak, üzerinde ihtiyaç duyduğu geliştirmeleri yaparak kendi konumsal portal uygulamalarında kullanabilecektir. Böylelikle, UKVA ortamında hem hızlı portal geliştirimi desteklenecek, hemde portal geliştirim maliyeti, düşürülecektir.

UKVA ortamında konumsal portlet katalođu oluşturulmasıyla ticari yazılım grupları, konumsal portallar için geliştirilen açık ve özgür kaynak kodlu uygulamaları geliştirerek, daha çok hizmeti bünyesinde barındıran konumsal portlet uygulamalarını enterprise edition olarak bu katalogdan yayınlayabilecektir. Bu yaklaşım, UKVA ortamında yeni bir iş sahası yaratması açısından önem arz eden bir yaklaşımdır.

4- SONUÇLAR

Konumsal portallar, KVAların gerçekleştirilmesi için önemli bir bileşendir. Bu nedenle dünya genelinde ülkeler çok sayıda konumsal portallar geliştirmişlerdir. Geliştirilen portallar incelendiğinde, her portal için benzer veya aynı uygulamalar tekrardan geliştirilmiştir. Bu nedenle, portal geliştiriminde zaman ve kaynak kaybı ortaya çıkmaktadır.

Bu tez çalışmasında, konumsal portalların gerçekleştirilmesine yönelik bir çerçeve, ortaya koyulmuştur. UKVA ortamında konumsal portalların nasıl geliştirilmesi gerektiği, ortaya koyulmuş ve bu doğrultuda prototip konumsal portallar geliştirilmiştir. Bu çerçeve, UKVA ortamında hızlı portal geliştirimi, portal ortamında web servisleri kompozisyonunun sağlanmasıyla uygulama tabanlı konumsal portal geliştirimi, UKVA ortamında açık ve özgür kaynak kodlu portal uygulamaları yaklaşımlarını, içermektedir.

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, Orman ve Su İşleri Bakanlığı gibi kurumlar 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 portal uygulamalarında, “yeniden kullanılabilirlik” ve konumsal portallar arasında “birlikte işlerlik” sorunu mevcuttur. Bu çalışma kapsamında konumsal portal geliştiriminde “yeniden kullanılabilirlik” ve “birlikte işlerlik” sorunlarına çözüm getiren bir yaklaşım dile getirilmiştir. Konumsal portallar için geliştirilen web uygulamalarının yeniden kullanılabilirliği ve birlikte işlerliği sorununa, ancak geliştirilen uygulamaların belli standartlara göre düzenlenmesiyle çözüm getirilebilecektir. Bu da ancak portlet mimarisi ile mümkündür.

Portletler için Java Portlet Belirtimi (JSR 286) ve WSRP standartları mevcuttur. Bu standartlara göre hazırlanan herhangi bir portlet, bu standartları sağlayan herhangi bir portal sunucusunda çalışacaktır. Böylelikle konumsal portallar için “*konumsal portletler*” geliştirerek, konumsal portallarda geliştirilen web uygulamalarının “yeniden kullanılabilirliği” sağlanabilecektir.

UKVA uygulamalarının gerisinde kalmış ülkelerin portallarını “hızlı” ve “kaliteli” bir şekilde geliştirebilmesi için mutlaka yeniden kullanılabilir portletlerin kullanılması gerekmektedir. Belli standartlara göre hazırlanmış konumsal portallar ve portletler, hızlı ve kaliteli bir portal geliştirimi için gereklidir. Ülkemizde, KVA çalışmalarında geride kalmış olmasından dolayı, hızlı portal geliştirmeye ihtiyaç duyulmaktadır. UKVA ortamında, konumsal portalları gerçekleştirmekte sorumlu kurum ve kuruluşlar portal oluşturulması sürecinde, bu gereklilikleri göz önünde bulundurmalıdırlar. Sonuç olarak, ilgili standartların kullanılması ile yeniden kullanılabilir portletler üretilebilir ve bunlar hızlı portal geliştirmeyi gerektiren ortamlarda kullanılabilir. Bu tez çalışmasında, “yeniden kullanılabilir” portletler ile konumsal portal geliştirmeye yönelik uygulama gerçekleştirilmiştir.

Bu çalışmada, 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. Bu tespitin detaylandırılması ve örneklendirilmesi, başka bir çalışmaya bırakılmıştır.

Bununla birlikte, W3C WSM nin tüm taraflar ya da kurumlarca henüz benimsenememiş olduğu günümüzde, WSK nun tümüyle W3C WSM ye bağlı kalarak gerçekleştirilmesi zaten mümkün olamamaktadır. Bu bakımdan portlet teknolojisi şu an en geçerli çözümü sunmaktadır. Bu, ülkemiz UKVA ortamı bakımından da son derece önemli bir sonuçtur. Çünkü mevcut durumda UKVA’ da yer alacak kurumlar, bir UKVA gerçekleştirimi için henüz hazır değildir. Ancak, UKVA’ ya olan ihtiyaç, kendini her geçen gün daha fazla hissettirmektedir. Bu durumda da, UKVA nın bir an önce başlatılarak, hızla gerçekleştirilmesi çok büyük bir önem arz etmektedir. Portal teknolojisi de bu bağlamda çok önemli bir katkı sunabilecek potansiyele sahiptir.

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 tez kapsamında konumsal portallar için getirilen bir diğer yaklaşım ise, açık ve özgür kaynak kodlu konumsal portal yaklaşımıdır. Bu çalışmada, bir konumsal portalın 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.

5. KAYNAKLAR

- Abdelnur, A. ve Hepper, S., 2003. Java Portlet Specification, Version 1.0, Status: FCS Specification, Specification Lead: Sun Microsystems, Inc., Release: August 29.
- Akıncı, H. ve Cömert, Ç., 2007. Ukva İçin Portal Teknolojisinin Değerlendirilmesi, TMMOB Harita ve Kadastro Mühendisleri Odası Ulusal Coğrafi Bilgi Sistemleri Kongresi, Ekim, Trabzon.
- 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, Trabzon.
- Akram, A., Chohan, D., Wang, X.D., Yang, X. and Allan, R., 2005. A Service Oriented Architecture for Portals Using Portlets. CCLRC e-Science Centre, CCLRC Daresbury Laboratory, Warrington WA4 4AD, UK.
- Arindam, D. and Ghosh, S. K., 2010. A Framework for Ubiquitous Geospatial Information Integration On Mobile Device Using Orchestration Of Geoservices, International Journal Of UbiComp (IJU).
- Ball, M., 2005. GeoSpatial One-Stop Moves to Phase Two Weaving a Tapestry of National Geospatial Data Coverage, GeoWorld, <http://www.geoplance.com/uploads/FeatureArticle0505sdi.asp>.
- Bostancı, H. T. ve Cömert, Ç., 2010. *Tapu Kadastro Web Servisleri İçin Tasarım Metodolojisi Geliştirilmesi*, HKM Jeodezi, Jeoinformasyon ve Arazi Yönetimi, 101-12, 63-68.
- Brauner J., Foerster T., Schaeffer B. and Baranski B., 2009. Towards a Research Agenda for Geoprocessing Services, 12th AGILE International Conference on Geographic Information Science, page 1 of 12, Leibniz Universität Hannover, Germany.
- Chung, L., Fang, Y., Chou, T., Lee, B. J. and Baranski, B., 2009. A SOA based debris flow monitoring system Architecture and proof-of-concept implementation, Held at Geoinformatics 2009, Washington, DC, USA.
- CIOC, Open source, open standards and re-use: Government action plan. London, The Cabinet Office. Available online at: http://www.cabinetoffice.gov.uk/media/123372/090224_opensource.pdf , Jan 27th 2010.
- 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, Istanbul, Turkey.
- De Jesus, J., Walker P. and Grant M., Groom S., 2011. WPS orchestration using the Taverna workbench: The eScience approach. Computers & Geosciences, doi:10.1016/j.cageo.

- Deegree WPS, 2009. Deegree Web Processing Service v2.5 manual, lat/lon GmbH, www.lat-lon.de
- Diaz, A. L., Fischer, P., Leue, C. and Schaeck, T., 2002. Web Services for Remote Portals (WSRP), Note 21 January.
- Díaz, L., Pepe, M., Granell, C., Carrara, P. and Rampini, A., 2010. Developing And Chaining Web Processing Services For Hydrological Models, Commission IV, 1st International Workshop on Pervasive Web Mapping, Geoprocessing and Services Conference.
- Díaz, L., Granell, C. and Gould, M., 2008. Case Study: Geospatial Processing Service for Web-based Hydrological Applications, Geospatial Services and Applications for the Internet, ISBN-13: 978-0-387-74673-9.
- Diaz, O. and Rodriguez, J.J., 2004. Portlets as Web Components: an Introduction, Journal of Universal Computer Science, 10, 4.
- Diaz, O., Iturrioz, J. and Irastorza, A., 2005 Improving Portlet interoperability through deep annotation, International worldwide Web Conference Committee (IW3C2), Chiba, Japan, 372-381.
- Fleuren, T. and Müller, P., 2008. BPEL Workflows Combining Standard OGC Web Services and Grid-enabled OGC Web Services, Software Engineering and Advanced Applications.
- Foust, J., Tang, W.S.M. and Selwood, J., 2005. Evolving Infrastructure: Growth and Evolution of Spatial Portals, From Pharaohs to Geoinformatics, FIG Working Week 2005 and GSDI-8, Cairo, Egypt April 16-21.
- GENESIS, 2011. The Genesis Solution Generic European Sustainable Information Space for Environment, <http://www.genesis-fp7.eu/>.
- Geoserver WPS. Geoserver WPS Manual, <http://docs.geoserver.org/stable/en/user/extensions/wps/processes.html>, 25.12.2011
- Granell, C., Díaz, L., Goulda, M., 2008. Geospatial Web Service Integration And Mashups For Water Resource Applications, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Volume: XXXVII, Beijing.
- Granell, C., Díaz, L., Gould, M., Pascual, V., Guimet, J., Carrara, P. and Pepe, M., 2007. Developing geoprocessing services for a hydrological model application, 27th EARSeL Symposium: Geoinformation in Europe, June, Bolzano.
- Hepper, S., 2008. Java Portlet Specification, Version 2.0, Status: FCS Specification, Specification Lead: Sun Microsystems, Inc., Release: January 25.

- Inspire, 2005. Inspire Infrastructure for Spatial Information in Europe, <http://inspire.jrc.ec.europa.eu/>.
- Kiehle, C., Greve, K. and Heier, C., 2007. Requirements for Next Generation Spatial Data Infrastructures-Standardized Web Based Geoprocessing and Web Service Orchestration, *Transactions in GIS*, 819–834.
- Lanig, S., Walenciak, G. and Zipf, A., 2011. Interoperable Processing of Sensor Data in SDIs – A Use Case for Wind Power Analysis, *ISW-2011: Integrating Sensor Web and Web-based Geoprocessing, An AGILE 2011 Conference Workshop*; Utrecht, The Netherlands, April 18.
- Linwood, J. and Minter, D., 2004. *Building Portals with the Java Portlet API*, ISBN (pbk): 1-59059-284-0.
- Maguire, D. J. and Longley, P. A., 2005. The Emergence of Geoportals and Their Role In Spatial Data Infrastructures, *Comput., Environ. and Urban Systems* 29, 3–14.
- Meng, X., Xie, Y., Bian, F., 2010. Distributed Geospatial Analysis through Web Processing Service: A Case Study of Earthquake Disaster Assessment, *Journal of Software*, ISSN 1796-217X, 5, 6.
- Nagappan, R., Skoczylas, R. and Sriganesh, R. P., 2003. *Developing Java Web Services, Concurrency And Computation: Practice And Experience*, *Concurrency Computat.: Pract. Exper.* 19, 1729–1738 Wiley Publishing Inc., Indiana, USA.
- North, C., 2006. *ESRI's Approach to Metadata and Portals*.
- OASIS, 2007. *OASIS Web Services Business Process Execution Language (WSBPEL) TC Committee Specification Version 2.0*.
- OASIS, 2008. *OASIS Web Services for Remote Portlets TC Committee Specification 2.0*.
- OGC, 2004a. *Geospatial Portal Reference Architecture, A Community Guide to Implementing Standards-Based Geospatial Portals, Version: 0.2*, OGC 04-039, OGC Discussion Paper.
- OGC, 2004b. *Web Map Service (WMS), Version 1.3.0*, OGC 04-024, OGC Implementation Specification.
- OGC, 2005a. *Web Feature Service (WFS) Implementation Specification, Version 1.1.0*, OGC 04-094, OGC Implementation Specification.
- OGC, 2005b. *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.
- OGC, 2010. *OpenGIS Web Coverage Service, Version: 2.0.0*, OGC 09-110r3, OGC Standard

- Oracle, 2008. Oracle® WebLogic Portal Federated Portals Guide 10g Release 3 (10.3), (2008).
- Osmond, M., Guo, Y., 2005. Adopting and Extending Portlet Technologies for e-Science Workflow Deployment, in Proc. UK e-Science All Hands Meeting 2005 (AHM'05), Nottingham, UK, Sept.
- Peltz, C., 2003. Web services orchestration A review of emerging technologies, tools, and standards, Hewlett Packard Company.
- PyWPS. PyWPS Manual, <http://pywps.wald.intevation.org/documentation/index.html>. 25.12.2010.
- Rao, R., 2009. Portal Server Development, Create dynamic, feature-rich, and robust enterprise portal applications.
- Rampini, A., De Michele, C., Lehning, M., Blöschl, G., Brilly, M., Lladós, A., Sapio, F. and Gould, M., 2006. AWARE: A tool for monitoring and forecasting Available Water REsource in mountain environment, Geophysical Research Abstracts, 8, 10780.
- Schäffer, B., 2010. Web Processing Service Study, BRISEIDE - BRIdgingServices, Information and Data for Europe ,WP1 - Inter-domain interoperability and user requirement analysis,T.1.5—Survey on existing Data and GI Services, D1.1.0.4—Survey on existing Data and GI Services.
- Stollberg, B. and 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.
- Stollberg, B. ve Zipf A., 2008. Geoprocessing Services for Spatial Decision Support in the Domain of Housing Market Analyses Experiences from Applying the OGC Web Processing Service Interface in Practice, 11th AGILE International Conference on Geographic Information Science 2008, 1-10, University of Girona, Spain
- Stollberg, B. ve Zipf, A., 2009. Development of a WPS Process Chaining Tool and Application in a Disaster Management Use Case for Urban Areas. UDMS 2009. 27th Urban Data Management Symposium, Ljubljana , Slovenia.
- Stollberg, B. ve Zipf, A., 2007. OGC Web Processing Service Interface for Web Service Orchestration - Aggregating Geo-processing Services in a Bomb Threat Scenario. In: Web and Wireless Geographical Information Systems. 7th International Symposium, W2GIS 2007, Cardiff, UK, November 2007 Proceedings. LNSC 4857, pp. 239-251. Springer-Verlag, Heidelberg.
- W3C, 2004. Web Services Architecture, WC3 Working Group Note, 11 February 2004. <http://www.w3.org/TR/ws-arch/>.
- Weiss, M., 2009. Portlet Standart JSR- 168-JSR 268, © Martin Weiss Informatik AG.

- Yang, X., Wangand, X. D. and Allan, R., 2006. Experiences with WSRP Testing and Development 18th January, Portsmouth, UK
- Yang, X., Wangand, X. D. and Allan, R., 2007. Investigation of WSRP support in selected open-source portal frameworks, Published online 7 June 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cpe.1116
- Yang, P., Evans, J., Cole, M., Marley, S., Alameh, N. and Bambacus, M., 2007. The Emerging Concepts and Applications of the Spatial Web Portal, American Society for Photogrammetry and Remote Sensing Vol. 73, No. 6, June 2007, pp. 691–698.
- Yildirim, M.E. ve Cömert, Ç., 2011. Portal Teknolojisi Ve Ulusal Konumsal Veri Altyapisi, TMMOB Harita ve Kadastro Mühendisleri Odası, 13. Türkiye Harita Bilimsel ve Teknik Kurultayı, Nisan 2011, Ankara.
- Yildirim, M.E. ve Cömert, Ç., 2011. Ulusal Konumsal Veri Altyapısında Portal Teknolojisi ve Servis Kompozisyonu, TMMOB Harita ve Kadastro Mühendisleri Odası, Coğrafi Bilgi Sistemleri Kongresi, Ekim, Antalya.
- Zoo WPS, 2011. Zoo WPS manual, <http://zoo-project.org/docs/index.html>.
- 52 North WPS, 2011. 52 North WPS Manual, <http://52north.org/communities/geoprocessing/wps/index.html>.
- URL-1, INSPIRE Geoportal, <http://www.inspire-geoportal.eu/> , 22.01.2012.
- URL-2, Geoportail France, <http://www.geoportail.fr/>, 22.01.2012.
- URL-3, IDEE portal, http://www.ideo.es/show.do?to=pideep_pidee.ES , 22.01.2012.
- URL-4, Go-Geo! Portal, <http://www.gogeo.ac.uk/cgi-bin/index.cgi> , 22.01.2012.
- URL-5, Portale Cartografico Nazionale, <http://www.pcn.minambiente.it/PCN/>, 22.01.2012.
- URL-6, Geospatial Data Service Centre Portal, <http://gdsc.nlr.nl/gdsc/en>, 22.01.2012.
- URL-7, GNII geoportal, <http://www.geoportal.lt>, 22.01.2012.
- URL-8, Geo.admin.ch, <http://www.geo.admin.ch/internet/geoportal/en/home.html>, 22.01.2012.
- URL-9, Liferay, Açık kaynak kodlu portal sunucusu, <http://www.liferay.com/>, 22.01.2012.
- URL-10, Exo Platform , Açık kaynak kodlu portal sunucusu, <http://www.exoplatform.com/company/public/website>, 22.01.2012.
- URL-11, StringBeans, Açık kaynak kodlu portal sunucusu, <http://sourceforge.net/projects/stringbeans/> , 22.01.2012.
- URL-12, uPortal, Açık kaynak kodlu portal sunucusu, <http://www.jasig.org/uportal>, 22.01.2012.

- URL-13, WSRP4J, Açık kaynak kodlu portal sunucusu, <http://portals.apache.org/wsrp4j/>, 22.01.2012.
- URL-14, WSDL sürümlerine göre döküman yapısı, http://en.wikipedia.org/wiki/File:WSDL_11vs20.png, 22.01.2012.
- URL-15, Liferay Portlet Kataloğu, <http://www.liferay.com/community/wiki/wiki/Main/Liferay+Portlets>, 22.01.2012.
- URL-16, SyncEX Portlet Kataloğu, http://www.syncex.com/portlet_catalog/portlet_catalog.htm , 22.01.2012.
- URL-17, IBM Lotus and WebSphere Portal Business Solutions Catalog, <https://greenhouse.lotus.com/>, 22.01.2012.
- URL-18, Jetspeed Portlet Kataloğu, <http://portals.apache.org/jetspeed-1/catalog.html>, 22.01.2012.
- URL-19, AWARE: A tool for monitoring and forecasting Available WATER REsource in mountain environment, URL 19 <http://www.aware-eu.info>, 22.01.2012.
- URL-20, Genesis Geoportal, Generic European Sustainable Information Space for Environment, <http://www.genesis-fp7.eu/> , 22.01.2012.
- URL-21, Web üzerinde google haritaları tabanlı web arayüzü geliştirimi için API, <http://code.google.com/intl/tr-TR/apis/maps/index.html>, 22.01.2012.
- URL- 22, Apache Struts, dinamik java web uygulamaları geliştirmek için bir çatı, <http://struts.apache.org/>, 22.01.2012.
- URL-23, Free and open source software, <http://en.wikipedia.org/wiki/FOSS> , 22.01.2012.
- URL-24, Özgür Yazılım Tanımı, <http://www.gnu.org/philosophy/free-sw.tr.html> , 22.01.2012.
- URL-25, Neden Açık Kaynak Özgür Yazılımın noktasını kaçırıyor, <http://www.gnu.org/philosophy/open-source-misses-the-point.tr.html> , 22.01.2012.
- URL-26, Why free software bettern than open source, <http://www.webcitation.org/5TchyyzYm> , 22.01.2012.
- URL-27, Free software vs Open Software, <http://cs-exhibitions.uni-klu.ac.at/index.php?id=224>, 22.01.2012.
- URL-28, Open Source Technology and Esri, <http://www.esri.com/news/arcnews/spring11/articles/open-source-technology-and-esri.html>, 22.01.2012.
- URL-29, Open Source Initiative, <http://opensource.org/> , 22.01.2012.

EKLER

EK 1. Portlet Modlari

```
package com.test;
import javax.portlet.GenericPortlet;
import javax.portlet.ActionRequest;
import javax.portlet.RenderRequest;
import javax.portlet.ActionResponse;
import javax.portlet.RenderResponse;
import javax.portlet.PortletException;
import java.io.IOException;
import javax.portlet.PortletRequestDispatcher;
/**
 * GoogleMapsViewerWithSpatialServices Portlet Class
 */
public class GoogleMapsViewerWithSpatialServices extends GenericPortlet {
    public void processAction(ActionRequest request, ActionResponse response) throws
    PortletException,IOException {
    }
    public void doView(RenderRequest request,RenderResponse response) throws
    PortletException,IOException {
        response.setContentType("text/html");
        PortletRequestDispatcher dispatcher =
        getPortletContext().getRequestDispatcher("/WEB-
        INF/jsp/GoogleMapsViewerWithSpatialServices_view.jsp");
        dispatcher.include(request, response);
    }
    public void doEdit(RenderRequest request,RenderResponse response) throws
    PortletException,IOException {
        response.setContentType("text/html");
        PortletRequestDispatcher dispatcher =
        getPortletContext().getRequestDispatcher("/WEB-
        INF/jsp/GoogleMapsViewerWithSpatialServices_edit.jsp");
        dispatcher.include(request, response);
    }
    public void doHelp(RenderRequest request, RenderResponse response) throws
    PortletException,IOException {
        response.setContentType("text/html");
        PortletRequestDispatcher dispatcher =
        getPortletContext().getRequestDispatcher("/WEB-
        INF/jsp/GoogleMapsViewerWithSpatialServices_help.jsp");
        dispatcher.include(request, response);
    }
}
```

EK 2 “Portlet Eventing” için Örnek Uygulama

Geocoder.java

```
package com.test;
```

```
import javax.portlet.GenericPortlet;
```

```
import javax.portlet.ActionRequest;
```

```
import javax.portlet.RenderRequest;
```

```
import javax.portlet.ActionResponse;
```

```
import javax.portlet.RenderResponse;
```

```
import javax.portlet.PortletException;
```

```
import java.io.IOException;
```

```
import javax.portlet.PortletRequestDispatcher;
```

```
import javax.xml.namespace.QName;
```

```
/**
```

```
 * Geocoder Portlet Class
```

```
 */
```

```
public class Geocoder extends GenericPortlet {
```

```
    public void processAction(ActionRequest request, ActionResponse response) throws
        PortletException,IOException {
```

```
        try {
```

```
            QName qName = new QName("http://sunevents.com", "geocoderevent", "x");
```

```
            com.test.eventingGeocode geocoderevent_Data = new com.test.eventingGeocode();
```

```
            geocoderevent_Data.setInputcoord(request.getParameter("inputcoord"));
```

```
            response.setEvent(qName, geocoderevent_Data);
```

```
        } catch (Exception e) {
```

```
            e.printStackTrace();
```

```
        } }
```

```
    public void doView(RenderRequest request,RenderResponse response) throws
```

```
        PortletException,IOException {
```

```
        response.setContentType("text/html");
```

```
        PortletRequestDispatcher dispatcher =
```

```
            getPortletContext().getRequestDispatcher("/WEB-INF/jsp/Geocoder_view.jsp");
```

```
        dispatcher.include(request, response);
```

```
    }
```

```
    public void doEdit(RenderRequest request,RenderResponse response) throws
```

```
        PortletException,IOException {
```

```
        response.setContentType("text/html");
```

```
        PortletRequestDispatcher dispatcher =
```

```
            getPortletContext().getRequestDispatcher("/WEB-INF/jsp/Geocoder_edit.jsp");
```

```
        dispatcher.include(request, response);
```

```
    }
```

Ek 2' nin devamı

```

public void doHelp(RenderRequest request, RenderResponse response) throws
PortletException,IOException {
response.setContentType("text/html");
PortletRequestDispatcher dispatcher =
getContext().getRequestDispatcher("/WEB-INF/jsp/Geocoder_help.jsp");
dispatcher.include(request, response);
}
}

```

GoogleMapViewwithSpatialServices.java

```

package com.test;
import javax.portlet.GenericPortlet;
import javax.portlet.ActionRequest;
import javax.portlet.RenderRequest;
import javax.portlet.ActionResponse;
import javax.portlet.RenderResponse;
import javax.portlet.PortletException;
import java.io.IOException;
import javax.portlet.PortletRequestDispatcher;
/**
 * GoogleMapViewwithSpatialServices Class
 */
public class GoogleMapViewwithSpatialServices extends GenericPortlet {
public void processAction(ActionRequest request, ActionResponse response) throws
PortletException,IOException {
}
public void doView(RenderRequest request,RenderResponse response) throws
PortletException,IOException {
response.setContentType("text/html");
PortletRequestDispatcher dispatcher =
getContext().getRequestDispatcher("/WEB-INF/jsp/Height_view.jsp");
dispatcher.include(request, response);
}
public void doEdit(RenderRequest request,RenderResponse response) throws
PortletException,IOException {
response.setContentType("text/html");
PortletRequestDispatcher dispatcher =
getContext().getRequestDispatcher("/WEB-INF/jsp/Height_edit.jsp");
dispatcher.include(request, response);
}
}

```


Ek 2' nin devamı

```

public void doHelp(RenderRequest request, RenderResponse response) throws
PortletException,IOException {
response.setContentType("text/html");
PortletRequestDispatcher dispatcher =
getContext().getRequestDispatcher("/WEB-INF/jsp/Height_help.jsp");
dispatcher.include(request, response);
}
@javax.portlet.ProcessEvent(qname = "{http://sunevents.com}geocoderevent")
public void handleProcessgeocodereventEvent(javax.portlet.EventRequest request,
javax.portlet.EventResponse response) throws javax.portlet.PortletException,
java.io.IOException {
javax.portlet.Event event = request.getEvent();
com.test.eventingGeocode value = (com.test.eventingGeocode) event.getValue();
String inputcoordinates = new String();
inputcoordinates = (String)value.getInputcoord();
response.setRenderParameter("inputcoordinates",inputcoordinates);
//response.setRenderParameter("key",value);
// request.getPortletSession().setAttribute("key", value);
}
}

```

eventingGeocode.java

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package com.test;
import java.io.Serializable;
/**
 *
 * @author emre
 */
public class eventingGeocode implements Serializable {
private String inputcoord;
public void setinputcoord(String inputcoord) {
this.inputcoord = inputcoord;
}
public String getinputcoord() {
return inputcoord;
}
}

```

Ek 2' nin devamı

Portlet.xml

```

<?xml version='1.0' encoding='UTF-8' ?>
<portlet-app xmlns='http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation='http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd
http://java.sun.com/xml/ns/portlet/portlet-app_2_0.xsd' version='2.0'>
<portlet>
<description>Geocoder</description>
<portlet-name>Geocoder</portlet-name>
<display-name>Geocoder</display-name>
<portlet-class>com.test.Geocoder</portlet-class>
<expiration-cache>0</expiration-cache>
<supports>
<mime-type>text/html</mime-type>
<portlet-mode>VIEW</portlet-mode>
<portlet-mode>EDIT</portlet-mode>
<portlet-mode>HELP</portlet-mode>
</supports>
<resource-bundle>com.test.messages</resource-bundle>
<portlet-info>
<title>Geocoder</title>
<short-title>Geocoder</short-title>
</portlet-info>
<supported-publishing-event xmlns:x='http://sunevents.com'>
<qname>x:geocoderevent</qname>
</supported-publishing-event>
</portlet>
<portlet>
<description>Height</description>
<portlet-name>Height</portlet-name>
<display-name>Height</display-name>
<portlet-class>com.test.Height</portlet-class>
<expiration-cache>0</expiration-cache>
<supports>
<mime-type>text/html</mime-type>
<portlet-mode>VIEW</portlet-mode>
<portlet-mode>EDIT</portlet-mode>
<portlet-mode>HELP</portlet-mode>
</supports>

```

Ek 2' nin devamı

```
<resource-bundle>com.test.messages</resource-bundle>
<portlet-info>
<title>Height</title>
<short-title>Height</short-title>
</portlet-info>
<supported-processing-event xmlns:x='http://sunevents.com'>
<qname>x:geocoderevent</qname>
</supported-processing-event>
</portlet>
<event-definition xmlns:x='http://sunevents.com'>
<qname>x:geocoderevent</qname>
<value-type>com.test.eventingGeocode</value-type>
</event-definition>
</portlet-app>
```

EK 3 örnek GetCapabilities dökümanı

```

<wps:Capabilities service="WPS" version="1.0.0" xml:lang="en"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsGetCapabilities_response.xsd"
xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:gml="http://www.opengis.net/gml"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"> <ows:ServiceIdentification> <ows:Title>deegree 3 WPS</ows:Title>
<ows:Abstract>deegree 3 WPS implementation</ows:Abstract>
<ows:ServiceType>WPS</ows:ServiceType>
<ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion> </ows:ServiceIdentification>
<ows:ServiceProvider> <ows:ProviderName>Mey KTU</ows:ProviderName>
<ows:ProviderSite xlink:href="http://www.harita.ktu.edu.tr"></ows:ProviderSite>
<ows:ServiceContact> <ows:IndividualName>Muhammet Emre
YILDIRIM</ows:IndividualName> <ows:PositionName>Developer</ows:PositionName>
<ows:ContactInfo> <ows:Phone> <ows:Voice>0462 377 27 66</ows:Voice>
<ows:Facsimile>0462 377 27 66</ows:Facsimile> </ows:Phone> <ows:Address>
<ows:DeliveryPoint>KTÜ Harita Mühendisliği Böl.</ows:DeliveryPoint>
<ows:City>Trabzon</ows:City> <ows:AdministrativeArea> </ows:AdministrativeArea>
<ows:PostalCode>61080</ows:PostalCode> <ows:Country>Turkey</ows:Country>
<ows:ElectronicMailAddress>muhammetemreyildirim@gmail.com</ows:ElectronicMailAddress
> </ows:Address>
</ows:ServiceContact> </ows:ServiceProvider> <ows:OperationsMetadata> <ows:Operation
name="GetCapabilities"> <ows:DCP> <ows:HTTP> <ows:Get
xlink:href="http://localhost:8080/deegree-wps-demo/services?"></ows:Get>

```

Ek 3' ün devamı

```

<ows:Post xlink:href="http://localhost:8080/deegree-wps-demo/services"></ows:Post>
</ows:HTTP> </ows:DCP> </ows:Operation> <ows:Operation name="DescribeProcess">
<ows:DCP> <ows:HTTP> <ows:Get xlink:href="http://localhost:8080/deegree-wps-
demo/services?"></ows:Get> <ows:Post xlink:href="http://localhost:8080/deegree-wps-
demo/services"></ows:Post> </ows:HTTP> </ows:DCP> </ows:Operation> <ows:Operation
name="Execute"> <ows:DCP> <ows:HTTP> <ows:Get
xlink:href="http://localhost:8080/deegree-wps-demo/services?"></ows:Get> <ows:Post
xlink:href="http://localhost:8080/deegree-wps-emo/services"></ows:Post> </ows:HTTP>
</ows:DCP> </ows:Operation> </ows:OperationsMetadata> <wps:ProcessOfferings>
<wps:Process wps:processVersion="1.0.0"> <ows:Identifier>bufferMEY</ows:Identifier>
<ows:Title>buffer analizi</ows:Title> <ows:Abstract> Deegree WPS sunucusu tarafından
sağlanan, buffer işlemi sadece GML 3.1.1 versiyonu ile çalışabilmektedir. Geliştirilen bufferMEY
işlemi ile GML'in 3.1.1, 2.1.2, 3.0.1, 3.2.1 versiyonlarına buffer uygulanabilmektedir. Ayrıca,
geliştirilen bu işlem ile bir WFS sunucusu tarafından gelen GML verisine buffer analizi
gerçekleştirilebilmektedir.</ows:Abstract> </wps:Process> <wps:Process
wps:processVersion="1.0.0"> <ows:Identifier>intersectMEY</ows:Identifier>
<ows:Title>Intersect analizi</ows:Title> <ows:Abstract> Deegree WPS sunucusu tarafından
sağlanan, intersect işlemi sadece GML 3.1.1 versiyonu ile çalışabilmektedir. Geliştirilen
intersectMEY işlemi ile GML'in 3.1.1, 2.1.2, 3.0.1, 3.2.1 versiyonları ile intersect analizi
gerçekleştirilebilmektedir. Ayrıca, geliştirilen bu işlem ile bir WFS sunucusu tarafından gelen
GML verileri ile intersect analizi gerçekleştirilebilmektedir. </ows:Abstract> </wps:Process>

```

Ek 3' ün devamı

```
<wps:Process wps:processVersion="1.0.0"> <ows:Identifier>unSuitZoneMEY</ows:Identifier>
<ows:Title>Unsuitzone</ows:Title>
```

```
<ows:Abstract>
```

Geliştirilen bu WPS işlemi, katı atık depolama tesisi yapılamayacak alanların tespitini gerçekleştirmektedir. GML'in 3.1.1, 2.1.2, 3.0.1, 3.2.1 versiyonları ile çalışabilmektedir. Bir WFS sunucusundan gelen GML verisi ile işlem yapabilmektedir.

```
</ows:Abstract> </wps:Process>
```

```
<wps:Process wps:processVersion="1.0.0"> <ows:Identifier>bufferIntRoadMEY</ows:Identifier>
```

```
<ows:Title>bufferIntRoadMEY</ows:Title> <ows:Abstract>
```

Geliştirilen bu WPS işlemi, kompozit WPS işlemine örnek olması bakımından geliştirilmiştir. Bu WPS işlemi, ilk önce yol verisine buffer uygulamakta ve sonuç katmanı ile kadastral katman kullanılarak intersect analizini gerçekleştirerek kamulaştırma yapılacak parsellerin tespiti işlemini gerçekleştirmektedir.

```
</ows:Abstract> </wps:Process>
```

```
</wps:ProcessOfferings> <wps:Languages> <wps:Default> <ows:Language>en</ows:Language>
```

```
</wps:Default> <wps:Supported> <ows:Language>en</ows:Language> </wps:Supported>
```

```
</wps:Languages> <wps:WSDL xlink:href="http://localhost:8080/deegree-wps-
```

```
demo/services?service=WPS&version=1.0.0&request=GetWPSWSDL"></wps:WSDL>
```

```
</wps:Capabilities>
```

Ek 4 Örnek DescribeProcess dökümanı

```

<wps:ProcessDescriptions service="WPS" version="1.0.0" xml:lang="en" xsi:schemaLocation="http://www.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsDescribeProcess_response.xsd" xmlns:wps="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:ogc="http://www.opengis.net/ogc" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ProcessDescription wps:processVersion="1.0.0" storeSupported="true" statusSupported="false">
    <ows:Identifier>BufferDENEME</ows:Identifier>
    <ows:Title>Process for buffering geometries contained in a GML stream.</ows:Title>
    <ows:Abstract>Processes a GML input stream by buffering every contained geometry element.</ows:Abstract>
    <DataInputs>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier>GMLInput</ows:Identifier>
        <ows:Title>GMLInput</ows:Title>
        <ComplexData>
          <Default>
            <Format>
<MimeType>text/xml; subtype=gml/3.1.1</MimeType>
</Format>
          </Default>
          <Supported>
            <Format>
              <MimeType>text/xml; subtype=gml/3.1.1</MimeType>
            </Format> <Format>
              <MimeType>text/xml; subtype=gml/3.2.1</MimeType>
            </Format>
          </Supported>
        </ComplexData>
      </Input>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier>BufferDistance</ows:Identifier>
        <ows:Title>Buffer distance</ows:Title>
        <LiteralData>
          <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-2/#double">double</ows:DataType>
          <UOMs>
            <Default>
              <ows:UOM>unity</ows:UOM>
            </Default>
          </UOMs>
          <ows:AnyValue></ows:AnyValue>
        </LiteralData>
      </Input>
    </DataInputs>
  </ProcessDescription>
</wps:ProcessDescriptions>

```

Ek 4' ün devamı

```
<ProcessOutputs>
  <Output>
    <ows:Identifier>BufferedGML</ows:Identifier>
    <ows:Title>BufferedGML</ows:Title>
    <ComplexOutput>
      <Default>
        <Format>
          <MimeType>text/xml</MimeType>
        </Format>
      </Default>
      <Supported>
        <Format>
          <MimeType>text/xml</MimeType>
        </Format>
      </Supported>
    </ComplexOutput>
  </Output>
</ProcessOutputs>
</ProcessDescription>
</wps:ProcessDescriptions>
```


Ek 5 Örnek Execute dökümanı

```

<wps:ExecuteResponse service="WPS" version="1.0.0" xml:lang="en" xsi:schemaLocation=
"http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsExecute_response.xsd" serviceInstance="http://local
host:8080/deegree-wps-
demo/services?service=WPS&request=GetCapabilities&version=1.0.0" xmlns:wps
="http://www.opengis.net/wps/1.0.0" xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:og
c="http://www.opengis.net/ogc" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="htt
p://www.w3.org/2001/XMLSchema-instance">
  <wps:Process wps:processVersion="1.0.0">
    <ows:Identifier>Buffer</ows:Identifier>
    <ows:Title>Process for buffering geometries contained in a GML stream.</ows:Title>
    <ows:Abstract>Processes a GML input stream by buffering every contained geometry
element.</ows:Abstract>
  </wps:Process>
  <wps:Status creationTime="2011-11-27T13:48:13.585Z">
    <wps:ProcessSucceeded>Process execution finished@2011-11-
27T13:48:13.583Z</wps:ProcessSucceeded>
  </wps:Status>
  <wps:ProcessOutputs>
    <wps:Output>
      <ows:Identifier>BufferedGML</ows:Identifier>
      <ows:Title>BufferedGML</ows:Title>
      <wps>Data>
        <wps:ComplexData mimeType="text/xml">
          <gml:Polygon srsName="EPSG:4326" xmlns:gml="http://www.opengis.net/gml">
            <gml:exterior>
              <gml:LinearRing srsName="EPSG:4326">
                <gml:posList>7.22000000 50.72000000 7.21807853 50.70049097
7.21238795 50.68173166 7.20314696 50.66444298 7.19071068 50.64928932 7.17555702
50.63685304 7.15826834 50.62761205 7.13950903 50.62192147 7.12000000 50.62000000
7.10049097 50.62192147 7.08173166 50.62761205 7.06444298 50.63685304 7.04928932
50.64928932 7.03685304 50.66444298 7.02761205 50.68173166 7.02192147 50.70049097
7.02000000 50.72000000 7.02192147 50.73950903 7.02761205 50.75826834 7.03685304
50.77555702 7.04928932 50.79071068 7.06444298 50.80314696 7.08173166 50.81238795
7.10049097 50.81807853 7.12000000 50.82000000 7.13950903 50.81807853 7.15826834
50.81238795 7.17555702 50.80314696 7.19071068 50.79071068 7.20314696 50.77555702
7.21238795 50.75826834 7.21807853 50.73950903 7.22000000 50.72000000</gml:posList>
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
        </wps:ComplexData>
      </wps>Data>
    </wps:Output>
  </wps:ProcessOutputs>
</wps:ExecuteResponse>

```

Ek 6 Örnek WSDL dokümanı

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<wsdl:definitions name="deegreeWPS"
targetNamespace="http://www.deegree.org/services/wps"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wps="http://www.deegree.org/services/wps"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/">
<wsdl:types>
<xsd:schema elementFormDefault="qualified"
targetNamespace="http://www.deegree.org/services/wps">
<xsd:element name="GetCapabilities">
<xsd:complexType>
<xsd:sequence>
<xsd:element maxOccurs="1" minOccurs="0" name="updateSequence"
type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="0" name="acceptFormats"
type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="0" name="acceptVersions"
type="xsd:string" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="GetCapabilitiesResponse">
<xsd:complexType>
<xsd:sequence>
<xsd:element maxOccurs="1" minOccurs="0"
name="GetCapabilitiesResult" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="DescribeProcess_ALL">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="in" type="xsd:string"></xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>

```

Ek 6'nin devamı

```

<xsd:element name="DescribeProcess_ALLResponse">
<xsd:complexType>
<xsd:sequence>
<xsd:element maxOccurs="1" minOccurs="0"
name="DescribeProcess_AllResult" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="DescribeProcess_ParameterDemoProcess">
<xsd:complexType>
<xsd:sequence>
<xsd:element maxOccurs="1" minOccurs="0" name="updateSequence"
type="xsd:string" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>

<xsd:element name="DescribeProcess_BufferProcess">
<xsd:complexType>
<xsd:sequence>
<xsd:element maxOccurs="1" minOccurs="0" name="updateSequence"
type="xsd:string" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="DescribeProcess_BufferProcessResponse">
<xsd:complexType>
<xsd:sequence>
<xsd:element maxOccurs="1" minOccurs="0"
name="DescribeProcess_BufferProcessResult" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="Execute_BufferProcess">
<xsd:complexType>
<xsd:sequence>
<xsd:element maxOccurs="1" minOccurs="0" name="GmlUrlResource"
type="xsd:string" />

```

Ek 6'nin devamı

```

</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="Execute_BufferProcessResponse">
<xsd:complexType>
<xsd:sequence>
<xsd:element maxOccurs="1" minOccurs="0"
name="ExecuteProcess_BufferProcessResult" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
</wsdl:types>
<wsdl:message name="GetCapabilitiesRequest">
<wsdl:part name="parameters" element="wps:GetCapabilities"></wsdl:part>
</wsdl:message>
<wsdl:message name="GetCapabilitiesResponse">
<wsdl:part name="parameters" element="wps:GetCapabilitiesResponse"></wsdl:part>
</wsdl:message>

<wsdl:message name="DescribeProcess_BufferProcessRequest">
<wsdl:part name="parameters" element="wps:DescribeProcess_BufferProcess"></wsdl:part>
</wsdl:message>
<wsdl:message name="DescribeProcess_BufferProcessResponse">
<wsdl:part name="parameters"
element="wps:DescribeProcess_BufferProcessResponse"></wsdl:part>
</wsdl:message>
<wsdl:message name="Execute_BufferProcessRequest">
<wsdl:part name="parameters" element="wps:Execute_BufferProcess"></wsdl:part>
</wsdl:message>
<wsdl:message name="Execute_BufferProcessResponse">
<wsdl:part name="parameters"
element="wps:Execute_BufferProcessResponse"></wsdl:part>
</wsdl:message>

<wsdl:portType name="deegree3WPS SOAP">
<wsdl:operation name="GetCapabilities">
<wsdl:documentation></wsdl:documentation>

```

Ek 6'nın devamı

```

<wsdl:input message="wps:GetCapabilitiesRequest" />
<wsdl:output message="wps:GetCapabilitiesResponse" />
</wsdl:operation>
<wsdl:operation name="DescribeProcess_BufferProcess">
<wsdl:documentation></wsdl:documentation>
<wsdl:input message="wps:DescribeProcess_BufferProcessRequest"></wsdl:input>
<wsdl:output message="wps:DescribeProcess_BufferProcessResponse"></wsdl:output>
</wsdl:operation>
<wsdl:operation name="Execute_BufferProcess">
<wsdl:documentation></wsdl:documentation>
<wsdl:input message="wps:Execute_BufferProcessRequest"></wsdl:input>
<wsdl:output message="wps:Execute_BufferProcessResponse"></wsdl:output>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="deegree3WPSSOAP" type="wps:deegree3WPSSOAP">
<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
<wsdl:operation name="GetCapabilities">
<soap:operation soapAction="http://www.deegree.org/services/wps/GetCapabilities"
style="document" />
<wsdl:input name="GetCapabilitiesRequest">
<soap:body use="literal" />
</wsdl:input>
<wsdl:output name="GetCapabilitiesResponse">
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="DescribeProcess_BufferProcess">
<soap:operation
soapAction="http://www.deegree.org/services/wps/DescribeProcess_BufferProcess"
style="document" />
<wsdl:input name="DescribeProcess_BufferProcessRequest">
<soap:body use="literal" />
</wsdl:input>
<wsdl:output name="DescribeProcess_BufferProcessResponse">
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="ExecuteProcess_BufferProcess">

```

Ek 6'nin devamı

```
<soap:operation
soapAction="http://www.deegree.org/services/wps/ExecuteProcess_BufferProcess"
style="document" />
<wsdl:input name="ExecuteProcess_BufferProcessRequest">
<soap:body use="literal" />
</wsdl:input>
<wsdl:output name="ExecuteProcess_BufferProcessResponse">
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="deegree3WPS">
<wsdl:documentation></wsdl:documentation>
<wsdl:port binding="wps:deegree3WPSSOAP" name="deegree3WPSSOAP">
<soap:address location="http://128.55.76.172:8080/deegree-wps-demo/services" />
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

ÖZGEÇMİŞ

1988 yılında Trabzon' da doğdu. İlk ve orta öğrenimini, Trabzon'da İsmetpaşa İlköğretim Okulunda tamamladı. Lise öğrenimini Trabzon'da Erdoğan lisesinde birincilikle tamamladı. 2004–2005 eğitim öğretim yılında Karadeniz Teknik Üniversitesi, Mühendislik Mimarlık Fakültesi, Jeodezi ve Fotogrametri Mühendisliği Bölümü'nü kazandı. 03.07.2009 tarihinde aynı bölümden mezun oldu. 2009–2010 öğretim yılında Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Harita Mühendisliği Anabilim dalında Yüksek lisans programına başladı. 2010 Ocak ayında Karadeniz Teknik Üniversitesi, Harita Mühendisliği bölümüne Araştırma Görevlisi olarak atandı. Halen bu görevine devam etmektedir. İngilizce bilmektedir.