

**KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

HARİTA MÜHENDİSLİĞİ ANABİLİM DALI

**KONUMSAL WEB SERVİSLERİNİN ÖZGÜR VE AÇIK KAYNAK KODLU
YAZILIMLARLA GERÇEKLEŞTİRİLMESİ**

YÜKSEK LİSANS TEZİ

Cemre YILMAZ

KASIM 2009

TRABZON

**KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

HARİTA MÜHENDİSLİĞİ ANABİLİM DALI

**KONUMSAL WEB SERVİSLERİNİN ÖZGÜR VE AÇIK KAYNAK KODLU
YAZILIMLARLA GERÇEKLEŞTİRİLMESİ**

Harita Müh. Cemre YILMAZ

**Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünde
“Harita Yüksek Mühendisi”
Unvanı Verilmesi İçin Kabul Edilen Tezdir.**

**Tezin Enstitüye Verildiği Tarih: 08.10.2009
Tezin Sözlü Savunma Tarihi : 19.11.2009**

**Tez Danışmanı : Doç. Dr. Çetin CÖMERT
Jüri Üyesi : Yrd. Doç. Dr. Oğuz GÜNGÖR
Jüri Üyesi : Yrd. Doç. Dr. Mustafa ULUTAŞ**

Enstitü Müdürü : Prof. Dr. Salih TERZİOĞLU

Trabzon 2009

ÖNSÖZ

“Açık Kaynak Kodlu Yazılımlarla Konumsal Web Servislerinin Gerçekleştirilmesi” konulu bu çalışma, Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Harita Mühendisliği Anabilim Dalında Yüksek Lisans Tezi olarak hazırlanmıştır.

Çalışma süresince bilgi, görüş ve önerilerinden yararlandığım değerli hocam Doç. Dr. Çetin CÖMERT’ e sonsuz teşekkürlerimi sunarım.

Yardımlarından dolayı değerli hocalarım Yrd. Doç. Dr. Halil AKINCI, Öğr. Gör. Hasan Tahsin BOSTANCI, Öğr. Gör. Bilgehan İMAMOĞLU’ na teşekkürlerimi sunarım.

Tez çalışmam süresince bilgi ve önerileriyle her zaman yanımda olan, yardımlarını hiçbir şekilde esirgemeyen değerli arkadaşlarım, Arş. Gör. Gülten KARA, Arş. Gör. Deniztan ULUTAŞ, Öğr. Gör. Emre İNCE, Yrd. Doç. Dr. Emine TANIR, , Arş. Gör. Leyla ÇAKIR’ a teşekkürlerimi sunarım.

Manevi desteklerinden dolayı değerli arkadaşlarım, Esra MULALİ, Şule CEYLAN, Yasemin SAĞDIÇ, Aydan YÜKSEL’ e sonsuz teşekkürlerimi sunarım.

Tüm hayatım boyunca olduğu gibi çalışmam sırasında da benden manevi desteklerini ve sevgilerini esirgemeyen anneme, babama ve kardeşlerime de teşekkürü büyük bir borç bilirim.

Cemre YILMAZ
Trabzon 2009

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ	II
İÇİNDEKİLER.....	III
ÖZET	IV
SUMMARY	VII
ŞEKİLLER DİZİNİ	VIII
TABLolar DİZİNİ.....	IX
SEMBOLLER DİZİNİ.....	X
1. GENEL BİLGİLER.....	1
1.1. Giriş.....	1
1.2. Web Servisleri.....	2
1.3. Özgür ve Açık Kaynak Kodlu Yazılımlar Hakkında Genel Bilgiler	3
1.3.1. Özgür Yazılım Vakfı (FSF)	3
1.3.2. Özgür Yazılım	4
1.3.3. Açık Kaynak Kodlu Yazılım	5
1.3.4. ÖAK Yazılımlarda Kullanılan Temel Lisanslar	5
1.3.4.1. GPL (General Public Licence)	5
1.3.4.1.1. İkili Lisanslama (Dual Licensing)	6
1.3.4.2. LGPL (Lesser General Public Licence)	8
1.3.5. ÖAK Yazılım Geliştirme Ortamı İçin Gereken Araçlar	8
1.3.6. Neden ÖAK Yazılım	9
1.3.7. ÖAK Yazılımlar ile Ticari Yazılımların İlişkileri	10
1.4. Açık Kaynak Kodlu GI İşleme Araçları ve Önemli Projeler	11
1.4.1. OGC	11
1.4.1.1. OGC Web Servisleri	11
1.4.1.2. OGC Web Processing Service (WPS)	12
1.4.1.2.1. WPS Operasyonları	12
1.4.1.3. OGC Simple Feature Specification for SQL (SFSQL)	13
1.4.1.4. OGC Filter Encoding Standard (FES)	13

1.4.2.	52° North	14
1.4.2.1.	52° North WPS	14
1.4.3.	Java Topology Suite	15
1.4.4.	GeoAPI	16
1.4.5.	GeoTools	17
1.4.6.	Sextante Projesi	22
1.4.7.	GRASS (Graphics Resources Analysis Support System)	24
1.4.8.	AWARE (A Tool For Monitoring and Forecasting Available WAter REsource in mountain Environments)	25
1.4.9.	GeOnAs (GeoBrain Online Analysis System).....	28
2.	YAPILAN ÇALIŞMALAR.....	32
2.1.	Örnek Uygulamanın ÖAK Yazılımlarla Gerçekleştirilmesi	32
2.1.1.	Örnek Uygulama İşlem Adımları	32
2.1.2.	Örnek Uygulama İşlem Adımlarının Ayrıntılı Açıklaması	32
2.1.2.1.	Uygulama İçin Gerekli Web Servislerinin Belirlenmesi	32
2.1.2.2.	Konumsal Web Servislerinin Gerçekleştirilmesinde Kullanılan ÖAK Yazılımın Belirlemesi	36
2.1.2.2.1.	Ön Seçim	36
2.1.2.2.1.1.	Literatür Taraması	36
2.1.2.2.1.2.	Genel ÖAK Yazılımları İçeren Web Siteleri	38
2.1.2.2.1.3.	Konumsal ÖAK Yazılımları İçeren Web Siteleri	42
2.1.2.2.2.	Nihai Seçim	45
2.1.2.2.2.1.	Uygulama Kapsamında İncelenen Yazılımlar	46
2.1.2.2.2.1.1.	Sextante	46
2.1.2.2.2.1.2.	GeOnAs GRASS Tabanlı Web Servisleri	48
2.1.2.2.2.1.3.	GeoTools	48
2.1.2.3.	Servis Kodlarının Geliştirilmesi ve Koşturulması	49
2.1.2.3.1.	Uygulamada Kullanılan Yazılımlar	49
2.1.2.3.1.1.	J2SDK (Java Software Development Kit)	49
2.1.2.3.1.2.	GeoTools Kütüphanesi	50
2.1.2.3.1.3.	Eclipse	50
2.1.2.3.1.4.	Yazılım Proje Yönetme Aracı.....	50
2.1.2.3.1.5.	Web Servisleri İçin Uygulama Sunucusu	52

2.1.2.3.2.	Uygulamada Geliştirilen Konumsal Web Servisleri	52
3.	BULGULAR VE İRDELEME	55
4.	SONUÇLAR VE ÖNERİLER	58
5.	KAYNAKLAR	59
6.	EKLER.	63
ÖZGEÇMİŞ		

ÖZET

Ülkemizde konumsal veri kullanan kurumlar, kamu kurumları, özel sektör ve yerel yönetimler arasında birlikte işlerliği sağlayacak bir altyapının bulunmaması sorunu hala devam etmektedir. Konumsal Veri Altyapıları (KVA) ile birlikte işlerliği sağlamak için KVA' ların gerçekleştirilmesinde önerilen teknoloji web servisleridir.

KVA' lar kapsamında üretilen işin ekonomik olmasının yanı sıra, KVA' ları oluşturan bileşenlerin de ekonomik ve hızlı üretimi büyük bir ihtiyaç durumundadır.

Bu çalışmada KVA' ları oluşturan web servislerinin Özgür Açık Kaynak (ÖAK) kodlu yazılımlarla gerçekleştirilmesi hedeflenmiştir. Bu amaçla çalışmada öncelikle konumsal ÖAK yazılımlar araştırılmıştır. Uygulama kapsamında belirlenen web servislerinin geliştirimi sırasında karşılaşılan sorunlar anlatılmıştır. Konumsal ÖAK yazılımların mevcut durumları belirlenerek, ticari yazılımlara göre avantajları ve dezavantajları ortaya koyulmuştur.

Anahtar Kelimeler: Özgür ve Açık Kaynak Kodlu Yazılımlar, Konumsal Web Servisleri

SUMMARY

Developing Spatial Web Services with Free and Open Source Software

In Turkey there does not exist any interoperability infrastructure amongst public and private sectors and local governments. The recommended technology is Web services to implement Spatial Data Infrastructures (SDI) for achieving interoperability with SDI.

It is important to produce work economically in scope of SDI. It is also very important that to produce the elements of SDI fast and economically.

In this study, the aim is at implementation of web services with Free and Open Source (FOS) software which constitute the SDI. Spatial FOS software are examined in this study. The problems that occur while implementing the web services for application are defined. Current status of spatial FOS software are specified, advantages and disadvantages have been determined according to commercial software.

Key Words: Free and Open Source Software, Spatial Web Services.

ŞEKİLLER DİZİNİ

	<u>Sayfa No</u>
Şekil 1. 52 North yazılımının genel lisans uygulaması	7
Şekil 2. 52 North yazılımının sıkı bağlı (tight-integration) ticari yazılımlarla lisans uygulaması.....	7
Şekil 3. 52 North yazılımının gevşek bağlı (loose coupling) ticari yazılımla lisanslanması.....	7
Şekil 4. 52 North yazılımının GPL lisanslı yazılım ile entegre edilmesi	8
Şekil 5. 52 North yazılımının geliştirilmesine paralel lisans uygulaması.....	8
Şekil 6. WPS operasyon şablonu	13
Şekil 7. 52 North WPS örnek operasyon şablonu “tampon bölge” işlemi	15
Şekil 8. 52 North WPS mimarisi	15
Şekil 9. Karşılaştırılan geometriler ile ikili karşılaştırmalar	16
Şekil 10. GeoTools Kütüphanesinin katmanları	19
Şekil 11. Sextante araç kutusu (toolbox).	23
Şekil 12. AWARE mimarisine genel bakış.....	26
Şekil 13. GeOnAs genel mimarisi.....	29
Şekil 14. GeOnAs online web servisleri kullanıcı ara yüzü	30
Şekil 15. Örnek uygulama krokisi	35
Şekil 16. Sourceforge.net sitesindeki “GIS” anahtar kelimesine göre arama sonuçlarını gösteren ara yüz.....	41
Şekil 17. Örnek konumsal ÖAK yazılımlar	44
Şekil 18. a) Parsel katmanı ve bina katmanı, b) 1. bölge, c) 2.bölge.....	47
Şekil 19. a) Sextante “clip” sonucu (1. bölge), b) Sextante “clip” sonucu (2. bölge)	47
Şekil 20. a) gvSIG “clip” sonucu (1.Bölge), b)gvSIG “clip” sonucu (2.Bölge).....	48
Şekil 21. http://sourceforge.net/projects/geotools/files/ ara yüzünün ekran görüntüsü....	51
Şekil 22. Eclipse Java IDE kullanıcı ara yüzü	52
Şekil 23. Eclipse uygulama geliştirme ara yüzü, geliştirilen web servis kodları.....	53
Şekil 24. ParselAlanHesapla kodunun sonuç ekran görüntüsü.....	54

TABLULAR DİZİNİ

	<u>Sayfa No</u>
Tablo 1. GeoTools katmanları ve görevleri	19
Tablo 2. GeoTools konumsal filtre ve karşılaştırma filtreleri tablosu.....	21
Tablo 3. GeoTools mantıksal filtreler.....	22
Tablo 4. AWARE WPS	27
Tablo 5. Grass tabanlı konumsal vektör harita işleme servisleri.....	31
Tablo 6. Konumsal web servisleri	33
Tablo 7. Konumsal olmayan web servisleri.....	34
Tablo 8. (Ramsey, 2007)'in ÖAK yazılımları uygulandıkları programlama dillerine ve kütüphane veya yazılım olmalarına göre yaptığı sınıflandırma	37
Tablo 9. (Ramsey, 2007)'in konumsal ÖAK web yazılımlarını türlerine göre yaptığı sınıflandırma	37
Tablo 10. (Steiniger, 2008)'in ÖAK yazılımları seçme ölçütleri.....	38
Tablo 11. Fsf.org sitesindeki yazılımların kategorileri.....	39
Tablo 12. Sourceforge.net sitesindeki yazılım kategorileri	40
Tablo 13. Osgeo.org sitesindeki yazılım kategorileri.....	42
Tablo 14. Maptools.org sitesinin yazılım kategorileri.....	43
Tablo 15. Freegis.org sitesindeki yazılım kategorileri	44

SEMBOLLER DİZİNİ

AWARE	: A tool for monitoring and forecasting Available Water Resource in mountain environments
CBS	: Coğrafi Bilgi Sistemleri
CQL	: Common Query Language
FES	: Filter Encoding Standard
FS	: Free Software (Özgür Yazılım)
GeOnAs	: GeoBrain Online Analysis System
GI	: Geographic Information
GNU	: GNU is Not Unix
GPL	: General Public Licence
GRASS	: Graphics Resources Analysis Support System
JDK	: Java Development Kit
JTS	: Java Topology Suite
KVA	: Konumsal Veri Altyapıları
LGPL	: Lesser General Public Licence
OGC	: Open Geospatial Consortium
OsGeo	: Open Source Geospatial Foundation
OSS	: Open Source Software
ÖAK	: Özgür ve Açık Kaynak
SFSQL	: Simple Feature Specification for SQL
SLD	: Styled Layer Descriptor
UKVA	: Ulusal Konumsal Veri Altyapıları
WPS	: Web processing Service

1. GENEL BİLGİLER

1.1. Giriş

Türkiye’de kamu kurumları, özel sektör, yerel yönetimler ve konumsal veri ile işlem yapan bütün kesimler arasında birlikte işlerliği “interoperability” sağlayacak bir alt yapı bulunmamaktadır. Birlikte işlerlik, donanım ve yazılım olarak farklı sistemlerin birlikte “iletişim kurabilmesi” olarak tanımlanabilir. Konumsal alanda birlikte işlerliği sağlayan altyapılar, Konumsal Veri Altyapıları (KVA) olarak adlandırılmaktadır. Çok daha eski bir kavram olan Ulusal Konumsal Veri Altyapısı (UKVA), ülke genelinde tüm kamu kurumları, özel sektör, yerel yönetimler, konumsal veri ile işlem yapan bütün kesimler arasında “birlikte işlerliği” sağlayacak ve vatandaşlar dahil ilgililere ihtiyaç duydukları veri ve servislere “anlık” erişim ve kullanım olanağı tanıyacak bir alt yapıdır (Akıncı, 2006). UKVA, katılımcı kurumlar arasında veri ve servislerin ortaklaşa kullanımına olanak tanıyarak hızlı, kaliteli ve ekonomik iş veya çözümler üretebilmeyi sağlayacaktır.

KVA’ların gerçekleştirilebilmesi için, önerilen teknoloji Web servisleridir ve KVA’ların Servis Yönelimli Mimari (SYM) ya da Web Servisleri (WS) yönelimli mimarinin birlikte işlerlik açısından nasıl gerçekleştirilebileceği (Akıncı, 2006)’da tanımlanmıştır.

UKVA kapsamında “üretilen işin” ekonomik yönünün önemli olmasının yanı sıra, bu alt yapıları oluşturan bileşenlerin de ekonomik ve hızlı üretimi bir ihtiyaç haline gelmiştir.

Konumsal alanda ülkemizde çoğunlukla ticari yazılımlar kullanılmaktadır. Bu yazılımların büyük bir kısmının “ithal yazılım” olarak ülkemizde sadece dağıtıcılıkları yapılmaktadır. Ticari yazılımları üreten firmaların talep ettikleri pahalı lisans ücretleri, bu yazılımların yaygın kullanımını olumsuz yönde etkilemektedir. Bu durumun sonucu olarak; kurumlar ihtiyaç duydukları konumsal (CBS) yazılımları ancak sınırlı sayıda alabilmekte, çoğu zamanda farklı kullanım amaçları olan yazılımları satın alma olanağı bulamamaktadırlar.

Son yirmi beş yıldır ortaya çıkan “özgür yazılım hareketi” yazılım geliştirme metodunu topluma yayma ve bu işi toplumsal bir üretim haline getirmeye çalışmaktadır. Bunun için en önemli gereksinim yazılımların kaynak kodlarının açık olarak dağıtılmasıdır. Kullanıcılar, bu sayede bu yazılımları istediği şekilde dağıtma, istediği

sayıda kopyalama, kod üzerinde deęişiklik yapma ve var olan kodlardan yeni yazılımlar üretme hakkına sahip olmaktadır. Bu durum Özgür ve Açık Kaynak kodlu (ÖAK) yazılım kullanıcılarına “teknik” ve “ekonomik” açıdan birçok avantaj sağlamaktadır. Örneęin, kullanıcılar ÖAK bir yazılımda uygulanmış bir rutini tekrar geliştirmek zorunda değillerdir ve uygulanmış olan rutinleri yeni çalışmaların içerisinde yeniden kullanabilmektedirler.

ÖAK yazılımların önemli bir dięer avantajı da çoęunlukla ücretsiz nadiren de düşük bir kullanım ücretiyle dağıtılmalarıdır.

Günümüzde birçok ülke yazılım geliştirme stratejisinde, sağladığı kullanıcı özgürlükleri, ekonomik katkı ve yazılım alanında dışa bağımlılıęın önlenmesi nedeniyle ÖAK yazılımların kullanılması yönünde kararlar almaktadır.

ÖAK yazılımlar, kullanıcılara birçok alanda çok çeşitli ürünler sunmaktadır. Bu alanlardan biri de Coęrafi Bilgi Sistemleri (CBS)’ dir. Günümüzde CBS alanında birçok ÖAK yazılım geliştirilmiştir. Bunlar, konumsal web sunucuları, masaüstü CBS programları, CBS kütüphaneleri vb. dir.

Çalışma kapsamında, konumsal alandaki ÖAK yazılımlar araştırılmıştır. KVA’ ların gerçekleştirilmesi açısından önemli bir yere sahip olan Web servislerinin, konumsal ÖAK yazılımlar kullanılarak gerçekleştirilmesi amaçlanmıştır.

1.2. Web Servisleri

W3C (2002), bir web servisini, İnternet tabanlı protokoller aracılığıyla XML tabanlı mesajları kullanarak dięer yazılım uygulamaları ile doğrudan etkileşimleri destekleyen, ara yüzleri ve bağlantıları XML tabanlı diller kullanılarak tanımlanabilen ve bulunabilen ve bir URI (Uniform Resource Identifier) tarafından tanımlanan bir yazılım uygulaması olarak tanımlamaktadır. Üst düzey bir görüşle bir Web servisi, belirli bir görevi gerçekleştirmek için internet üzerinden çağrılabilen bir uygulama olarak tanımlanabilir (Ulutaş, 2008).

1.3. Özgür ve Açık Kaynak Kodlu Yazılımlar Hakkında Genel Bilgiler

1.3.1. Özgür Yazılım Vakfı (FSF)

Özgür yazılım hareketi, Richard M. Stallman tarafından Unix işletim sisteminin yerini alabilecek yeni bir işletim sistemi olan GNU (GNU is Not Unix) projesiyle birlikte 1983 yılında başlamıştır.

Özgür Yazılım Vakfı, 1985 yılında merkezi Boston, ABD olmak üzere, dünya genelinde, bilgisayar kullanıcılarını özgürlüğe teşvik etmek ve tüm özgür yazılım kullanıcılarının haklarını savunma amacıyla kurulmuştur.

FSF, özgür yazılım kavramını tanımlamaktadır. Tamamen özgür yazılım olarak lisanslanması amaçlanan GNU işletim sistemini desteklemektedir. Bunun haricinde önemli diğer özgür yazılım projelerini de desteklemektedir. GNU yazılım idame ettiricileri için e-posta listeleri ve çatı (shell) servisleri içeren yazılım geliştirme sistemi sağlamaktadır. GNU işletim sistemine gönüllülerin katılımlarını kolaylaştırmak için kaynak kod deposu ve özgür yazılım geliştirme merkezi olan Savannah' ı sunmaktadır.

Günümüzde birçok özgür yazılım bulunmaktadır. GNU gibi büyük çapta işletim sistemi yazılımlarından birçok alanda yazılmış 5000'in üzerinde özgür program FSF/UNESCO özgür yazılım dizininde (<http://directory.fsf.org/>) listelenmektedir.

FSF, GNU işletim sisteminin ve özgür yazılımların büyük bir kısmının telif haklarını elinde bulundurmaktadır. Bunun amacı, özgür yazılımların tescilli (proprietary) yazılımlara dönüştürülmesini önlemektir. FSF, telif hakkı için yapılan başvuruları US copyright Office ile kayıt ederek, özgür yazılımlar için genellikle, GPL lisansını uygulamaktadır.

FSF' nin yayımladığı en önemli lisans, GPL (GNU General Public Licence) lisansıdır. FSF'nin yayımladığı diğer önemli lisanslar; GNU LGPL (GNU Lesser General Public License) , GNU AGPL the GNU Affero General Public License ve GNU FDL (GNU Free Document Licence) lisanslarıdır (URL-1, 2009). Özgür yazılım alanında en sık kullanılan lisanslar GPL ve LGPL lisanslarıdır. Bu nedenle bu iki lisans ayrıntılı olarak anlatılmıştır.

1.3.2. Özgür Yazılım (FS)

FSF' nin kurucusu Richard Stallman'e göre özgür yazılımın temelinde; kullanıcıların yazılımı kullanma, kopyalama, dağıtma, geliştirme, yazılım üzerinde çalışma ve değişiklik yapma özgürlüğü bulunmaktadır. Program kullanıcılarının dört temel özgürlüğünün olması gerekmektedir;

- Herhangi bir amaç için programı çalıştırma özgürlüğü (Özgürlük 0).
- Programın nasıl çalıştığını inceleme ve istekleri doğrultusunda değiştirme özgürlüğü (Özgürlük 1). Bu özgürlük için kaynak koda erişim önkoşuldur.
- Programın kopyalarını yeniden dağıtarak, diğer kullanıcılarla paylaşma özgürlüğü (Özgürlük 2).
- Programı geliştirme ve geliştirmeleri yayımlama (genel olarak değiştirilmiş sürümleri) böylece diğer kullanıcıların yararlanmasını sağlamak özgürlüğü (Özgürlük 3). Bu özgürlük için kaynak koda erişim önkoşuldur.

Kullanıcılar bu dört temel özgürlüğe sahipse, program özgür bir programdır. Bu nedenle, kullanıcı kopyalarını değiştirerek veya değiştirmeden, ücretsiz veya ücret karşılığında, kişi ve yer farkı gözetmeksizin dağıtmada özgür olmalıdır.

Özgür olmaktan kasıt kişinin belirtilenleri yaparken izin isteme veya ücret ödeme zorunluluğunun olmamasıdır.

Kullanıcı değişiklik yapabilmeli ve bunları kişisel çalışmalarında, değişiklik yaptığını belirtmeksizin kullanabilmelidir. Eğer değişikliklerini yayınlamışsa, hiç kimseyi hiçbir şekilde bu konuda özel olarak bilgilendirme gerekliliği olmamalıdır (URL-2, 2009).

Linux, ÖAK yazılımların en öne çıkan örneklerinden birisidir. Tüm kaynak kodu tamamıyla kullanılabilir, ücretsizce değiştirilebilir ve herhangi biri tarafından yeniden dağıtılabılır. Günümüzde neredeyse her ülke kendine ait Linux tabanlı işletim sistemini geliştirmiştir. Bu bağlamda, Pardus TÜBİTAK tarafından desteklenen ve Türkiye'de geliştirilen Linux tabanlı işletim sistemidir.

Özet olarak özgür yazılımda temel şart, yazılımı dağıtma ve geliştirme özgürlüğünün kimsenin tekeline bırakılmamasıdır.

1.3.3. Açık Kaynak Kodlu Yazılım

Açık kaynak kodlu yazılım fikri Açık Kaynak Girişimi (OSI) tarafından, Kaliforniya’da 1998 yılında ilk olarak özgür yazılım kavramındaki yanlış anlaşılmayı önlemek için ortaya atılmıştır. Özgür yazılım kavramındaki “özgür” kelimesi ile , “ücretsiz yazılım” değil “kullanıcılara belli özgürlükler veren yazılım” kastedilmektedir.

Açık kaynak savunucularının bir kısmı “açık kaynak” kavramını, “ özgür yazılım için pazarlama kampanyası” olarak nitelendirmiştir. Diğer bir kısmı ise, özgür yazılım fikrinin savunduğu etik değerleri reddetmiştir. Her iki grup ta, “açık kaynak” kavramından bahsederken, özgür yazılımın vurguladığı özgürlüklerin değil, yazılımın nitelikli ve kaliteli olması üzerinde durmuşlardır.

Sonuç olarak, açık kaynak kodlu yazılım kavramı, bir yazılımın kaliteli ve güvenilir bir şekilde geliştirilmesi ilkesiyle birlikte anılmaya başlanmıştır. Özgür yazılım ise, sosyal bir harekettir ve kullanıcıların yazılımı kullanma özgürlükleri düşüncesini vurgulamaktadır (URL-3, 2009).

Bir yazılımın veya kütüphanenin yukarıda bahsedilen “açık kaynak” veya “özgür yazılım” kavramlarından herhangi birine uygun olmasının, teknik olarak gelişiminde önemi yoktur. Bunun sebebi her iki durumda da esas olan yaklaşımın, yazılımın kaynak kodunun açık olarak dağıtılmasını gerektirmesidir.

Bu çalışma kapsamında literatürde de en sık kullanılan şekliyle; özgür yazılım ve açık kaynak kodlu yazılımı bir arada belirten “Özgür ve Açık Kaynak” (ÖAK) yazılım ifadesi kullanılacaktır.

1.3.4. ÖAK Yazılımlarında Kullanılan Temel Lisanslar

1.3.4.1. GPL (General Public Licence)

GPL lisans anlaşması, 1983 yılında Richard Stallman tarafından geliştirilmiştir. GPL, özgür yazılımları değiştirme ve paylaşma hakkının korunması ve yazılımın bütün kullanıcıları için serbest olması amacı ile yazılmıştır.

Kullanıcıların bu haklarının korunması iki aşamada gerçekleşmektedir:

- Yazılıma telif hakkı alınarak,

- Yazılım lisansı olarak kullanıcılara hukuki olarak, yazılımı kopyalama, dağıtma ve/veya değiştirme hakkı tanıyan GPL lisansı sunulmaktadır.

Yani, bir yazılımı GPL altında yayınlamak, yazara telif hakkı güvencesi altında yazılımının başkaları tarafından özgür yazılım felsefesine aykırı olarak dağıtılamayacağı güvencesini verir.

GPL, özellikle programın kaynak kodunun açık olmasının gerekliliğini savunmaktadır. Temel amaç, şeffaflık, kullanıcının ne kullandığını bilmesi, yazılımın çok kişi tarafından hızlıca geliştirilmesidir. GPL lisansına göre, kullanıcı program üzerinde değişiklik yaparak, programı yeniden dağıtma hakkına sahiptir. Fakat bunu yapmasının tek şartı, yayımladığı programın da GPL lisanslı olmasıdır (URL-4, 2009).

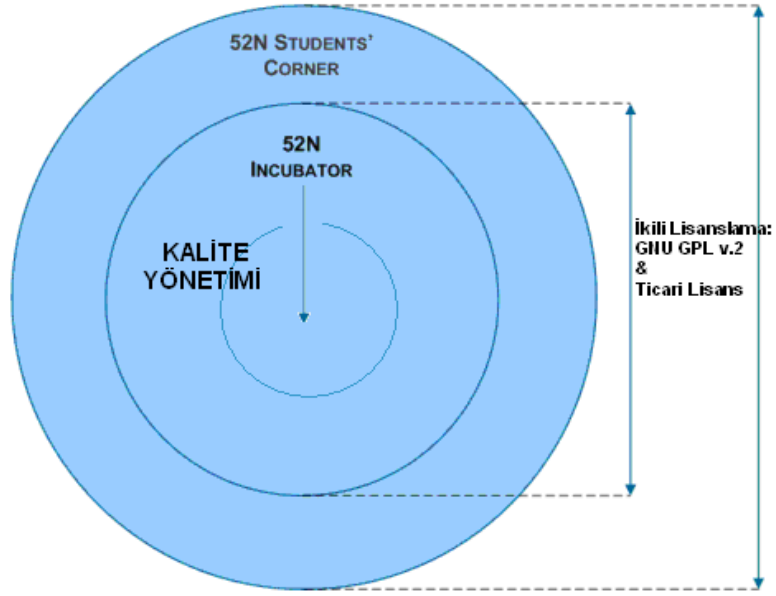
GPL lisanslı yazılımları ticari yazılımlarla kullanmak için ikili lisanslama yöntemi uygulanmaktadır.

1.3.4.1.1. İkili Lisanslama (Dual Licensing)

Bir yazılım aynı zamanda hem GPL lisansı ile hem de ticari bir lisansla lisanslanabilir. Bu yaklaşım ikili lisanslama olarak adlandırılmaktadır. Bu lisanslama yöntemi, ÖAK yazılımlara ticari yazılımlarla birlikte de kullanılabilme olanağı vermektedir. Bu yöntemle yazılım, kullanım ve dağıtımına göre, kaynak kodları aynı olmak koşuluyla iki farklı ürün olarak sunulmaktadır.

Örneğin, 52° North yazılımlarının lisanslanmasında bu yöntemi tercih etmektedir. 52°North GNU GPL v.2 lisansı Şekil 1 ile yazılımlarının özgür bir şekilde kopyalanmasını, dağıtılmasını ve değiştirilmesini sağlamaktadır. GPL lisanslı bir yazılımla birlikte kullanılmasına veya bağlanmasına izin vermektedir.

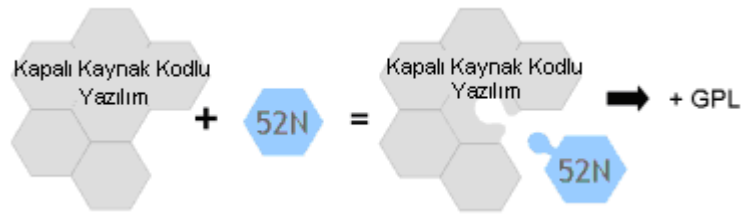
Ticari lisansla birlikte, 52° North' e bağlı veya 52° North yazılımını aday kod deposu (incubator) veya çekirdek kodundan kullanan yazılımların dağıtılmasına imkân tanımaktadır. Kaynak kodunu yayınlamak istemeyen kişiler veya yasal kuruluşlar için ve yazılımı 52°North yazılımı olmaksızın çalışmayan, sıkı-bağlı (tight-integration), kapalı kaynak kodlu yazılımlar için ticari lisans gerekmektedir (Şekil 2), (URL-5, 2009).



Şekil 1. 52°North yazılımının genel lisans uygulaması (URL- 6).

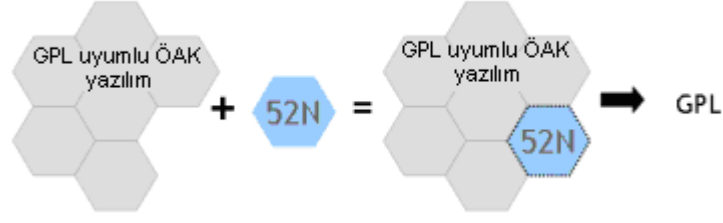


Şekil 2. 52°North yazılımının sıkı bağlı (tight-integration) ticari yazılımlarla Lisans uygulaması (URL- 6).

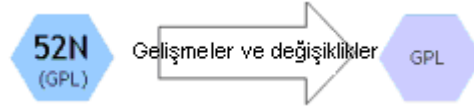


Şekil 3. 52°North yazılımının gevşek bağlı (loose coupling) ticari yazılımla lisanslanması (URL- 6).

Eğer yazılım GNU GPL uyumlu ise ticari lisansa gerek duyulmamaktadır Şekil 4-5;



Şekil 4. 52°North yazılımının GPL lisanslı yazılım ile entegre edilmesi (URL-6).



Şekil 5. 52°North yazılımının geliştirilmesine paralel lisans uygulaması (URL-6).

1.3.4.2. LGPL (Lesser General Public Licence)

GPL lisansından en önemli farkı, LGPL lisansını kullanan yazılımın, farklı bir lisansa sahip yazılım ile birlikte kullanılabilmesidir. Bu sayede GPL lisansının özgür yazılımlara getirdiği zorunluluk ortadan kalkmaktadır.

LGPL lisansı genel olarak yazılım kütüphaneleri için kullanılmaktadır. Bazı kütüphaneler ticari yazılımlar ile birlikte kullanılarak daha fazla tanınmak, bazı alanlarda “de-facto” standart haline gelmek isteyebilirler. Bu durum GPL lisansı ile kısıtlanmıştır.

Bu nedenle ticari bir yazılıma entegre edilmeleri için kütüphanelere bu hakkı veren bir lisans gerekmektedir. LGPL lisansı ile açık kaynak kodlu yazılımlar ticari yazılımlarla birlikte kullanılabilirler. Fakat ticari yazılıma gömülü olan kısmın yine LGPL lisanslı ve yine açık kaynaklı olarak yayınlanma zorunluluğu bulunmaktadır (URL- 7, 2009). LGPL lisanslı kütüphanelere örnek olarak, GeoTools kütüphanesi verilebilir.

1.3.5. ÖAK Yazılım Geliştirme Ortamı İçin Gereken Araçlar

ÖAK yazılım projelerinin geliştirilme işinin teknik yolunu kolaylaştırmak ve projelerin devamlılığını sağlamak için yazılım geliştirme ortamına ihtiyaç bulunmaktadır. Bu kapsamda ÖAK yazılım projesinin geliştirilmesi için gerekli ortak özellikler aşağıda açıklanmıştır;

Sürüm Kontrol Sistemi: ÖAK proje geliştiricilerinin zaman ve mekân birlikteliği sağlanamadığından, farklı yerlerdeki geliştiricilerin aynı kod üzerinde çalışabilmesi, yazılımın kaynak kodunda birçok geliştirici tarafından yapılan değişikliklerin izlenebilmesi, sürüm takibinin yapılabilmesi için sürüm kontrol sistemleri kullanılmaktadır. Subversion (SVN) ve Concurrent Version Systems (CVS) örnek olarak verilebilir.

Hata Takip ve Raporlama Sistemleri: ÖAK yazılımlarının kodların sabit olabilmesi için hataların bildirilmesi ve raporlanması büyük önem taşımaktadır. Bu nedenle ÖAK yazılımlar hata raporlama ve takip sistemleri kullanılmaktadır. Mozillanın gelişmiş bir sistemi olan Bugzilla ve Mantis örnek olarak verilebilir.

İletişim Kanalları: ÖAK yazılım geliştiricilerinin, kullanıcılarının birbirleriyle iletişim kurması, yazılım hakkında geri bildirimlerin ve yeniliklerin iletilmesinde önemli bir rol oynamaktadır. En çok tercih edilen iletişim yolu e-posta listeleri ve IRC (Internet Relay Chat) programlarıdır. Özellikle e-posta listeleri kullanıcılar için yazılım hakkında araştırma yapabilecekleri çok önemli bilgi kaynaklarıdır. E-posta listelerine örnek olarak, GNU Mailman ve IRC programlarına örnek olarak, xchat verilebilir.

Paket Hazırlama Sistemleri: ÖAK yazılımları paketlemeye yarayan sistemlerdir. Red Hat Package Manager (RPM) örnek olarak verilebilir.

Doküman Hazırlama: Bir yazılımın kullanıcı dostu kabul edilebilmesinin en önemli ölçütlerinden biri de kullanıcılar için iyi bir dokümantasyon sağlamasıdır. Bu nedenle ÖAK Yazılımlar için doküman hazırlama araçları (Latex, Lyx vb.) kullanılmaktadır (Kepoğlu ve Düzgün, 2008).

Bütün bu yazılımları ve gerekli olan diğerlerini web üzerinden sunan depolar bulunmaktadır. GNU Savannah, SourceForge, OSOR EU bunlara örnek olarak gösterilebilir.

1.3.6. Neden ÖAK Yazılım

Bilgi teknolojileri artık birçok ülkede kalkınmaya yönelik stratejik planlarda yer almaya başlamıştır. Her ülke bu konuda kendi stratejisini belirlemektedir. Bu alanda Çin, Almanya ve Brezilya kalkınma programlarında yazılımın ve bilgi teknolojilerinin önemini aldıkları kararlarla belirtmişlerdir ve ÖAK yazılımların teşvik edilmesini ve desteklenmesini ulusal bir politika olarak benimsemişlerdir.

ÖAK yazılımlar teşvik edilerek ve desteklenerek, yazılım konusunda ulusal alanda yazılımlar üretilerek, yazılım ithalatının önüne geçilebilmektedir.

Tescilli yazılımlar, standartlar konusunda her alanda kendi formatlarını dayatmaktadırlar. Bu da kurumlar arasında veya kurumların kendi bünyelerinde kullanılan yazılımların birbirleriyle çalışmasını engellemektedir. Oysa ÖAK yazılımlarla sağlanan kaynak kodlara sahip olma, bilginin toplumsal mülkiyeti, farklı sistemlerin birbirleriyle ilişkiye geçebilmesi için yapılacak düzenlemelere olanak sağlamaktadır (Gözükeleş, 2004).

Bu politikaları takip etmekle birlikte ÖAK yazılımlara yapılan teşviklerle birçok alanda ticari yazılımlara alternatif olabilecek ÖAK yazılımlar üretilmiştir. Örneğin, OpenOffice yazılımı MS Office yazılımının yerine geçebilecek olgunluğa erişmiş bir projedir. MS Office program formatlarıyla işlem yapabilmesi bu alanda kullanımının yaygınlaşması için önemli bir avantajdır. Kamu kurumlarında ve okullarda OpenOffice kullanımının yaygınlaştırılmasıyla, kurumlar yazılımın lisans ücreti yükünden kurtulmuş olacaklardır.

1.3.7. ÖAK Yazılımlar ile Ticari Yazılımların İlişkisi

ÖAK yazılımların kullanıcılara ulaşması ve kendisini geliştirmesi, gördüğü destekle doğru orantılıdır. ÖAK yazılımlara en büyük destek ticari firmalar ve gönüllüler tarafından yapılmaktadır. Bunun dışında varlığını sürdürebilmesi için farklı gelir kaynakları da sağlayabilmektedir. Örneğin, CD ortamında dağıttığı yazılımlardan ortam ücreti talep edebilmekte veya kullanıcılara servis hizmeti sağlayarak varlığını sürdürebilecek kazancı elde edebilmektedir. ÖAK yazılımlarını desteklemek ticari firmalara birçok fayda sağlamaktadır.

ÖAK yazılımlara katkıda bulunmak ticari firmalara birçok kullanıcıya ismini duyurma imkânı sağlamaktadır. Örneğin, Sun Microsystems firması, OpenOffice yazılımına verdiği destekle uluslararası büyüklükte kullanıcı kitlesine ismini duyurmaktadır.

Dünyanın her bölgesinden katılımcıların bir araya geldiği ÖAK toplulukları birçok nitelikli geliştiriciyi bünyesinde barındırır. Ticari firmalar, bu topluluklara verdikleri destek sayesinde geliştiricilerle yakın temas kurar. Bu sırada yüksek bilgi ve beceri potansiyeline sahip geliştiricileri, kendi şirketlerinin birer parçası haline getirebilirler. Bu

da firmaların yaratıcılık seviyelerini ve kalitelerini yükseltir. Böylece rakip firmalara karşı üstünlük sağlarlar.

ÖAK yazılımı destekleyen ticari firmalar rakiplerine karşı ücretsiz veya daha uygun ücrete sahip alternatif ve daha nitelikli yazılımlar geliştirilmesini sağlayarak rakiplerinin güç kaybına sebep olur.

Örneğin, Google firmasının özgür bir yazılım olan web tarayıcısı, Firefox' a verdiği destek, Microsoft firmasının geliştirmiş olduğu web tarayıcısı Internet Explorer'ın kullanım oranını ciddi ölçüde azaltmıştır. Dünyanın her bir yanındaki geliştiricileri sayesinde Firefox doğrudan kullanıcılarının ihtiyaçlarına cevap verebilmektedir. Bu durum Internet Explorer için geçerli değildir. Google verdiği bu destek sayesinde, dünya çapında yazılımcıların yaptığı çalışmaları yakından takip edebilmekte ve Firefox için geliştirdiği uygulamalarla dünya çapında kullanıcılar ile buluşarak kullanıcı kitlesini ve birinci derecede gelir kaynağını oluşturan reklam gelirlerini arttırmaktadır.

Ticari firmalar özgür yazılım ortamının gelişmesine katkı sağlamak amacı ile yazılımcıların eğitimlerinden, farklı platformlarda organize edilmelerinden vb. maddi kazanç sağlayabilir.

1.4. Açık Kaynak Kodlu GI İşleme Yazılımları ve Önemli Projeler

1.4.1. OGC

1994 yılında kurulan Open Geospatial Consortium (OGC), Coğrafi Bilgi Sistemleri (CBS) şirketlerinden, kamu kurumlarından ve üniversitelerden oluşan, yaklaşık 400 üyeye sahip, kar amacı gütmeyen uluslar arası bir endüstri birliğidir.

Birliğin amacı, mevcut CBS sistemleri arasındaki birlikte işlerlik (interoperability) problemlerini belirlemek ve bilgi teknolojilerindeki gelişmeleri yakından izleyerek bu problemleri giderecek, herkesin kullanımına açık ara yüz belirtileri geliştirmektir.

1.4.1.1. OGC Web Servisleri

OGC Web servisleri, konumsal veri sunmak için geliştirilmiş “insan-odaklı” servislerdir. Bir OGC Web servisi tarafından sunulan konumsal verileri elde etmek isteyen

bir kullanıcının, ilgili servis tarafından gerçekleştirilen bir dizi operasyonu kullanması gerekmektedir. Web Map Service (WMS), Web Feature Service (WFS) ve Web Processing Service (WPS) OGC Web servislerinden birkaçıdır. WMS, kullanıcıya raster formatında harita döndüren bir servistir, WFS ise kullanıcıya OGC'nin standart veri belirtimi olan Geography Markup Language (GML) formatında vektör veri döndürür. WPS servis standardı ve çalışma prensibi aşağıda detaylı olarak anlatılmıştır.

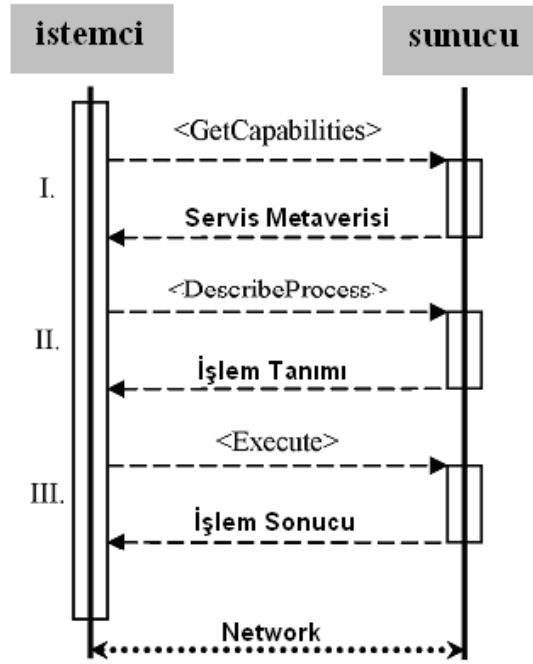
1.4.1.2. OGC Web Processing Service (WPS)

OpenGIS WPS ara yüz standardı, konumsal işlem servislerinin istek ve cevaplarını standartlaştırmak amacıyla belirli kurallar sağlamaktadır. WPS, ihtiyaç duyduğu veriye ağ üzerinden veya sunucudan erişebilir ve veri olarak, raster veri formatlarını kullanabileceği gibi, GML gibi veri dönüşüm standartlarını da kullanabilir. WPS' in sunacağı işlemler çok basit atomik işlemlerin yanı sıra çok karmaşık modeller de olabilir (OGC, 2005).

1.4.1.2.1. WPS Operasyonları

Şekil 6'da WPS servisinin çalışma prensibi gösterilmektedir. WPS'in kullandığı operasyonlar aşağıda detaylı olarak açıklanmıştır.

1. GetCapabilities (zorunlu): Bu operasyon kullanıcıya, WPS servisinin metaverisi ile WPS'in sağladığı işlemlerin isimleri ve genel olarak tanımlarını içeren XML (veya Capabilities) dokümanını döndürür.
2. DescribeProcess (zorunlu): Bu operasyon istemciye, WPS' de koştan işlemler hakkında gerekli girdiler, desteklediği formatlar ve üretilen çıktılar ile ilgili detaylı bilgi verir.
3. Execute (zorunlu): Bu operasyon kullanıcının, WPS tarafından uygulanan belirli bir işlemi; servis girdilerini kullanarak ve üretilen çıktıları geri döndürerek koşturmasını sağlar.



Şekil 6. WPS operasyon şablonu (OGC,2005)

1.4.1.3. OGC Simple Feature Specification for SQL (SFSQL)

Simple Feature Interface Standard (SFS), ilişkisel ve nesnel –ilişkisel veritabanlarında, coğrafi nesnelere erişmek için iyi belirtilmiş (well-defined) ve ortak bir yol sağlar.

OpenGIS Simple Features (SF); nokta, çizgi ve poligon vektör veri elemanlarıyla tanımlanan coğrafi nesnelere dir. SFS standardı iki bölümden oluşmaktadır.

Birinci bölüm, “ortak mimari”, SF veri saklama ve veri erişim ara yüzlerini kullanacak uygulamalar için ortak coğrafi nesne modeli sağlamaktadır.

İkinci bölüm, SFSQL, SF için standart bir SQL uygulaması sağlamaktadır (OGC, 2006).

1.4.1.4. OGC Filter Encoding Standard (FES)

FES, filtre ifadeleri için XML bir kodlama tanımlar. Bir filtre ifadesi, bir nesnenin özellikleri üzerinde kısıtlamalar kurar. Böylece üzerinde işlem yapılacak belirli bir grup nesne belirlenir. Kısıtlamalar konumsal, zamansal ve sayısal değerler üzerinde belirtilebilir. (OGC, 2005).

1.4.2. 52° North

52° North Initiative for Geospatial Open Source Software GmbH, IfGI (Institute for Geoinformatics of the University of Muenster), con Terra GmbH, ITC (International Institute for Geo-Information Science and Earth Observation) ve ESRI (Environmental Systems Research Institute Inc.) tarafından 2006 yılında kurulmuştur (URL-8, 2009).

52° North Initiative for Geospatial Open Source Software GmbH; araştırma, eğitim, öğretim ve kullanım için ÖAK coğrafi yazılımlar fikrini teşvik eden, geliştirilmesini, uygulanmasını kendine görev olarak üstlenen uluslar arası araştırma ve geliştirme kuruluşudur.

52° North, araştırma kuruluşları ve uluslararası CBS alanında önde gelen kişiler tarafından yönlendirilen açık girişimi destekler.

Bu kuruluşun araştırma ve geliştirme alanındaki işbirliği ortakları çalışmalarını; Sensor Web Enablement (SWE), Web Security ve Geo-Rights Management ve Geo-Processing alanlarında sürdürmektedirler (URL-8, 2009).

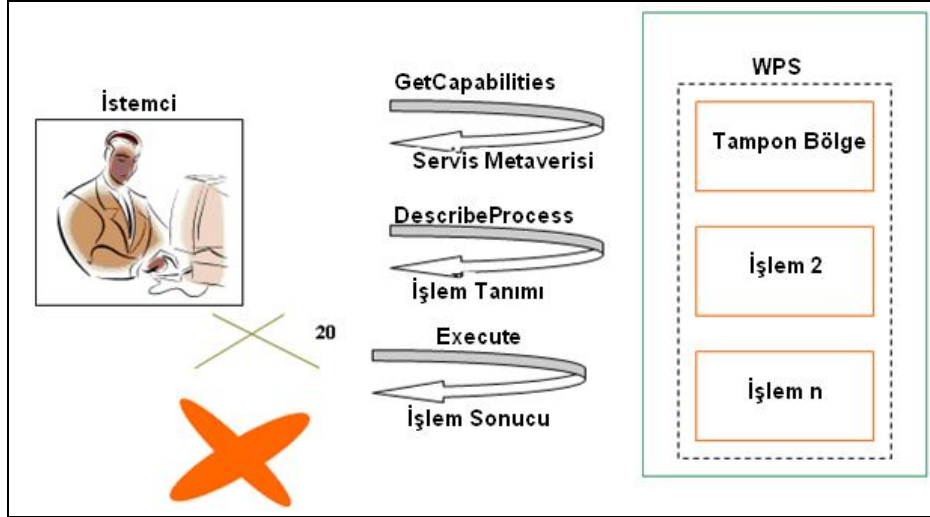
52°North yazılımları GPL lisansı altında yayınlanmaktadır. Yazılımları, ticari yazılımlarına entegre etmek isteyenler için aynı zamanda ikili lisans modeli uygulanmaktadır.

1.4.2.1. 52° North WPS

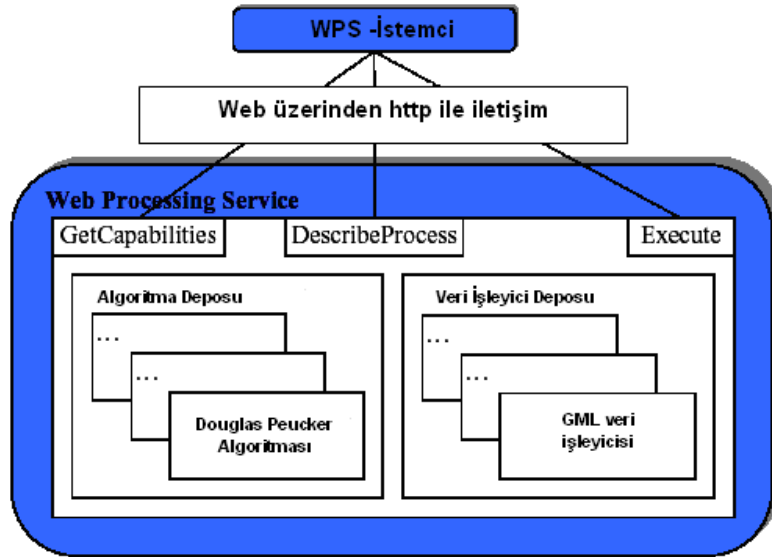
52° North WPS, OGC WPS 1.0.0 belirtiminin açık kaynak kodlu Java tabanlı bir uygulamasıdır. Güçlü kütüphaneler üzerine kurulmuştur (GeoTools, JTS, xmlBeans, derby, Servlet API). İşlemler ve XML veri işleme için eklenebilir (pluggable) bir çatı sunarak kullanıcılara 52N WPS'e kendi rutinlerini entegre etme olanağı sağlamaktadır. Yani kullanıcıya uygulamasını bir WPS servisiyle sunacak yapıyı sağlamaktadır.

52° North WPS, veri olarak, ArcGrid, GeoTIFF ve GML2.0 formatlarını desteklemektedir. Ayrıca, Sextante projesi kapsamında geliştirilen 220'den fazla rutin 52N WPS çatısına entegre edilerek kullanıma hazır halde bulunmaktadır (URL -9, 2009).

Şekil 7'de örnek "tampon bölge" işlemi gösterilmiştir. Şekil 8'de 52N WPS mimarisi gösterilmiştir.



Şekil 7. 52°North WPS örnek operasyon şablonu “tampon bölge” işlemi



Şekil 8. 52°North WPS mimarisi (Lemmens vd., 2007).

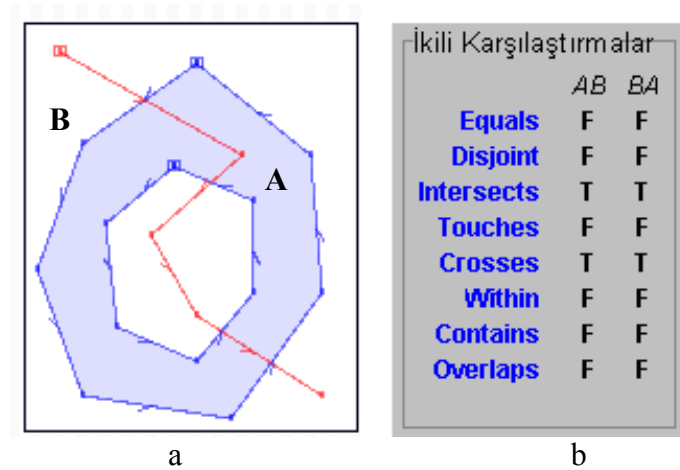
1.4.3. Java Topology Suite

Java Topology Suite (JTS), OGC Simple Feature Specification for SQL (SFS) standardında tanımlanan konumsal veri modelini uygulayan; Java tabanlı ve açık kaynak kodlu bir uygulama programlama ara yüzüdür (API).

JTS' yi önemli kılan ikili karşılaştırma (binary predicate) ve konumsal analiz metotları gibi iki boyutlu konumsal algoritmaları içermesidir. İkili karşılaştırma metotları

(eşittir, kesişir, içerir vb.); argüman olarak iki geometri olarak (Şekil 9.a), aralarındaki ilişkiyi belirlenen metoda göre karşılaştırır ve sonucu “boolean” bir değer (“true” veya “false”) olarak belirtir (Şekil 9.b). Konumsal analiz metotları (tampon bölge, kesişim, fark vb.), argüman olarak bir veya daha fazla geometri alır ve belirtilen metoda göre oluşturduğu yeni geometriyi döndürür.

JTS kütüphanesi diğer ÖAK (GeoTools, JUMP, gvSIG vb.) projeler tarafından kullanılmaktadır. JTS; GeoConnections, British Columbia Ministry of Sustainable Resource Management (MSRM), Centre for Topographic Information – Sherbrooke (CTIS) kurumlarının ortak proje sponsorluğunda geliştirilmiştir. Yazılımı geliştirme anlaşması, Vividsolutions, Inc.’e verilmiştir. Yazılım, LGPL lisansı altında dağıtılmaktadır (URL-10, 2009).



Şekil 9. Karşılaştırılan geometriler ile ikili karşılaştırmalar

1.4.4. GeoAPI

GeoAPI projesi, 2002 yılında GeoTools ve Open JUMP gibi bağımsız projeler için, ortak bir Java tabanlı uygulama programlama ara yüzü sağlamak amacıyla geliştirilmeye başlanmıştır. OGC de benzer amaçla, Geographic Objects (GO-1) projesini Polexis firmasına yaptırmaya başlamıştır. Bu iki proje benzer amaçlarla yapıldıkları için bir süre sonra birleştirilmiştir. OGC, Polexis firması tarafından tamamlanan, GeoAPI tabanlı Geographic Objects uygulama belirtimini 2005 yılında resmi olarak yayınlamıştır. Polexis firması farklı bir firma tarafından satın alındığı için projeye devam edilmemiştir.

Günümüzde, GeoAPI projesinin ana katılımcısı, GeoTools'tur. OGC'nin resmi olarak kabul ettiği sürümü GeoAPI 2.0' dır. GeoTools, GeoAPI 2.1' i GeoTools 2.4 sürümü için ve GeoAPI 2.2'yi ise GeoTools 2.5 sürümü için geliştirmiştir. Fakat bu projeler çalışma grubu kurulmadığı için sadece aşama (milestone) olarak yayınlanmaktadır (URL-11, 2009).

Günümüzde GeoAPI, konumsal uygulamalar için java dilinde geliştirilmiş ara yüzler sağlamaktadır. GeoAPI, meta veri kullanma, jeodezik referanslandırma, projeksiyon ve dönüşüm, referanslandırılmış görüntülerin kullanılması, vektör geometrinin kurulması ve işlenmesi, topolojik veri yapıları, coğrafi nesnelere tanımlanması ve kullanılması işlemlerini yapmak için OGC ve ISO standartlarını uygulayan çekirdek bir ara yüz kümesi oluşturmuştur. Bunun yanı sıra, veriye erişmek ve saklamak için filtre sorgularını içeren ara yüzler geliştirmiştir.

GeoAPI' nin bazı kısımlarını uygulayan projeler; GeoToolkit, GeoTools, GeoOxygene, Gnis, Jscience' dır (URL-12, 2009). GeoAPI ara yüzlerinin telif hakkı OGC' ye aittir.

1.4.5. GeoTools

GeoTools kütüphanesinin ilk sürümü 1996 yılında Leeds Üniversitesinde bir yüksek lisans öğrencisinin yüksek lisans tezi kapsamında, son kullanıcıların harita çizmeleri amaçlanarak geliştirilmeye başlanmıştır. GeoTools ikinci sürümünden itibaren genel kullanıcılardan ziyade, geliştiricilere (developers) yönelik tasarlanmaya başlanmıştır. GeoTools2, resmi olarak 2002 yılında www.geotools.org sitesinde duyurulmuştur (Hall ve Leahy, 2008).

Şu anda ikinci sürümünde, birçok ÖAK coğrafi yazılıma taban oluşturmaktadır. Bu yazılımlardan birkaçı; GeoServer, uDig, gvSIG, GeoVISTA studio ve JUMP' dır (Hall ve Leahy, 2008).

GeoTools günümüzde, CBS uygulamaları için açık kaynak kodlu bir Java kütüphanesidir. Diğer açık kaynak kodlu kütüphanelerden JTS ve GeoAPI'yi kullanmaktadır (Garnett, 2008).

JTS GeoTools'a, geometrik analizler yapabilme olanağı sağlar. İki coğrafi objenin kesişiminin veya farkının hesaplanması, geometrik analize örnek olarak gösterilebilir (Garnett, 2008).

GeoAPI, OGC Filter Encoding Specification v 1.1. standardını uygulamaktadır ve GeoTools'a coğrafi objelerden belirli özellikteki objeleri seçme ve bu objelerle işlem yapma olanağı sağlamaktadır. GeoTools'daki filtreler; konumsal filtreler, karşılaştırma filtreleri ve mantıksal filtrelerdir (Hall ve Leahy, 2008). GeoTools'daki filtreler Tablo 2-3'te detaylı olarak gösterilmiştir. Soğuksu mahallesindeki 2 katlı binaların bulunması filtreye örnek olarak gösterilebilir.

GeoAPI, konumsal referanslandırma için OpenGIS Spatial Referencing by Coordinates belirtimini uygulamaktadır ve GeoTools'a konumsal referanslandırma işlemlerini yapma olanağı sağlamaktadır. GeoTools farklı koordinat referans sistemlerini eklentiler (plugins) ile (epsg-hsql, epsg-access, epsg-wkt, epsg-postgresql) kullanmaktadır (Garnett, 2007).

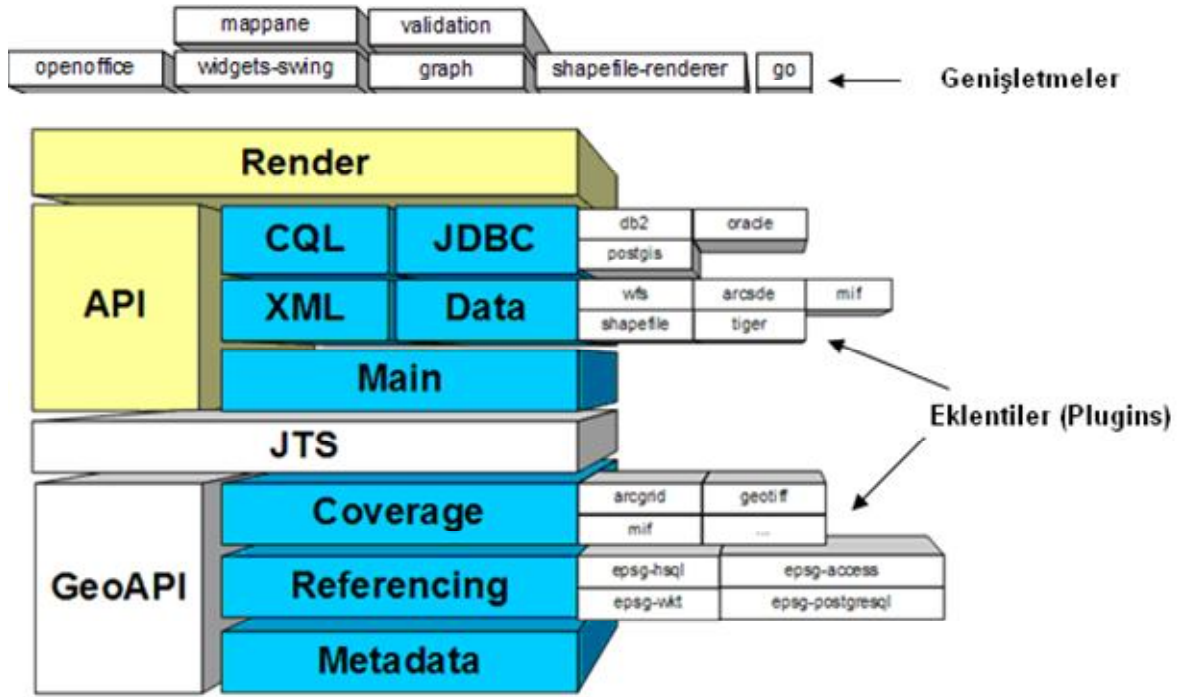
GeoTools farklı veri kaynaklarıyla uygulama geliştirmeye imkân tanır. GeoTools farklı veri kaynaklarıyla çalışmak için eklentiler kullanmaktadır. Kullandığı eklentiler; PostGIS, db2, oracle, shapefile, ArcSDE, GeoTIFF, ArcGrid, MIG'dir. Bu pluginler ayrıca Şekil 11'de gösterilmiştir.

GeoTools' un desteklediği raster veri kaynakları; WMS, ArcGrid, GeoTIFF ve world dosyasıyla referanslandırılmış görüntülerdir.

GeoTools'un desteklediği vektör veri kaynakları; shapefile, GML, WFS, PostGIS, OracleSpatial, ArcSDE, MySQL'dir (Garnett, 2007).

GeoTools konumsal uygulamalar dışında, kütüphaneye ek özellikler sağlayan genişletmeler içermektedir. Bu genişletmeler Şekil 10'da gösterilmiştir; MapPane, harita gösterimi için basit bir Java swing aracıdır. Shapefile- renderer, sadece "shapefile" için hızlı bir render aracıdır (Garnett, 2007).

Şekil 10'da GeoTools kütüphanesinin katmanları, Tablo 1'de bu katmanların görevleri kısaca açıklanmıştır.



Şekil 10. GeoTools kütüphanesinin katmanları (URL- 13, 2009).

Tablo 1. GeoTools katmanları ve görevleri (URL- 13, 2009).

Katman	Jar	Görevi
Render	gt2-render	Harita çizmek (Sadece 2D)
JDBC	gt2-jdbc	Veritabanından konumsal veri okumak
Data	gt2-data	Konumsal veri okumak
Main	gt2-main	Filtre, coğrafi objeleri kullanmak vb.
Coverage	gt2-coverage	Raster veri okumak
API	gt2-api	Koordinat dönüşümü, referencedenvelope vb.
Referencing	gt2-referencing	Referans koordinat sistemini tanımlamak
Metadata	gt2-metadata	Konumsal veriyi tanımlamak (ISO 19115)

GeoTools kütüphanesinde ki her katman bir alttakinin üzerine kuruludur (Şekil 11). Örneğin; “referencing” modülüyle çalışmak için; geoAPI, gt2-metadata ve gt2-referencing modüllerine ihtiyaç duyulur (Garnett, 2007). GeoTools, GeoAPI arayüzü aracılığıyla kullandığı filtrelerin yanı sıra OGC Catalog belirtimi tarafından tanımlanan CQL’i de (Common Query Language) desteklemektedir.

Haritanın Çizilmesi (Rendering): Bu modül henüz tamamlanamamıştır. GeoTools bu modülünde sembollendirme için OGC SLD (Styled Layer Descriptor) standardını uygulamaktadır. Şu anda, SLD 1.0 belirtimini desteklemektedir. Sadece 2D harita çizimi yapabilmektedir ve modülün geliştirilmesi için diğer projelerle (OpenJUMP, uDig, DeeGree) iş birliği yapmaya çalışılmaktadır (Garnett ve Bedward, 2006).

GeoTools sürekli gelişen bir kütüphanedir ve son olarak 2.6.0 sürümü duyurulmuştur. GeoTools kütüphanesi, LGPL lisansı altında yayınlanmaktadır (Roldan ve Garnett, 2008).

GeoTools' un desteklediği filtre örnekleri:

Karşılaştırma filtresine “propertyIsLessThan” örnek verecek olursak; “Derinlik” özniteliği 30'dan büyük olan nesnelerin sınıflandırılması verilebilir.

Konumsal filtre için, özellikleri belirtilen saran poligonun (boundingbox) sınırları içine düşen nesnelerin bulunması örnek olarak verilebilir.

Mantıksal filtreye örnek olarak, örnek verdiğimiz ilk iki filtreyi birleştirebiliriz. Bu durumda, belirli bir saran poligonun içerisine düşen derinlik özniteliği 30'dan büyük nesnelerin bulunması mantıksal filtreler ile gerçekleştirilir.

Tablo 2. GeoTools konumsal filtre ve karşılaştırma filtreleri tablosu (Hall ve Leahy, 2008)

Konumsal Filtreler	Geometrik argümanlar arasında belirtilen konumsal ilişkinin olup olmadığını kontrol eder. Doğru (True) veya Yanlış (False) değeri döndürür.
Bbox	Geometrinin, belirtilen “bounding box” ın içerisinde olup olmadığını kontrol eder.
Ötesinde	İlk geometrinin, diğer geometriye verilen bir mesafede olup olmadığını kontrol eder.
Kapsar	İlk geometrinin ikinci geometriyi kapsayıp kapsamadığını kontrol eder.
Keser	İlk geometrinin ikinci geometriyi kesip kesmediğini kontrol eder
Ayrık	İlk geometrinin ikinci geometriden ayrı olup olmadığını kontrol eder.
İçinde	İlk geometrinin herhangi bir parçasının verilen bir mesafeye göre ikinci geometrinin içerisinde olup olmadığını kontrol eder.
Eşittir	İki geometrinin eşit olup olmadığını kontrol eder.
Kesişir	İki geometrinin kesişip kesişmediğini kontrol eder.
Çakışır	İlk geometrinin iç bölgesinin ikinci geometrinin iç bölgesiyle çakışıp çakışmadığını kontrol eder.
Özellik	Coğrafi nesnenin özelliğine ait değeri getirir.
Temas eder	Coğrafi nesnenin geometrisinin, ikinci geometriye temas edip etmediğini kontrol eder .
İçerisinde	İlk geometrinin tam olarak ikinci geometrinin içerisinde olup olmadığını kontrol eder.
Karşılaştırma Filtreleri	Coğrafi nesnelerin özniteliklerini, belirtilen ifadelerle karşılaştırıp doğru olup olmadığını kontrol eder. Doğru veya yanlış değeri döndürür.
PropertyIsEqualTo	Her iki ifadenin (expression) birbirine eşit olup olmadığını kontrol eder.
PropertyIsGreaterThan	İlk ifadenin ikincisinden büyük olup olmadığını kontrol eder.
PropertyIsGreaterThanOrEqualTo	İlk ifadenin ikinci ifadeye eşit veya büyük olup olmadığını kontrol eder.
PropertyIsLessThan	İlk ifadenin ikinci ifadeden küçük olup olmadığını kontrol eder.
PropertyIsLessThanOrEqualTo	İlk ifadenin ikinci ifadeden küçük veya eşit olup olmadığını kontrol eder.
PropertyIsLike
PropertyIsNull	Bir ifadenin değerinin olup olmadığını kontrol eder.
PropertyIsBetween	Bir ifadenin değerinin verilen aralıklarda olup olmadığını kontrol eder.

Tablo 3. GeoTools mantıksal filtreler (Hall ve Leahy, 2008)

Mantıksal Filtreler	Farklı filtrelerin birleştirilmesini sağlar.
Ve	Her iki filtreyi birleştirir ve her iki filtre de “Doğru” sonucu veriyorsa “Doğru” olarak değer döndürür.
Veya	Birleştirilen filtrelerden herhangi biri “doğru” ise “doğru” olarak değer döndürür.
Not	İfadenin mantıksal değerini tersine döndürür.

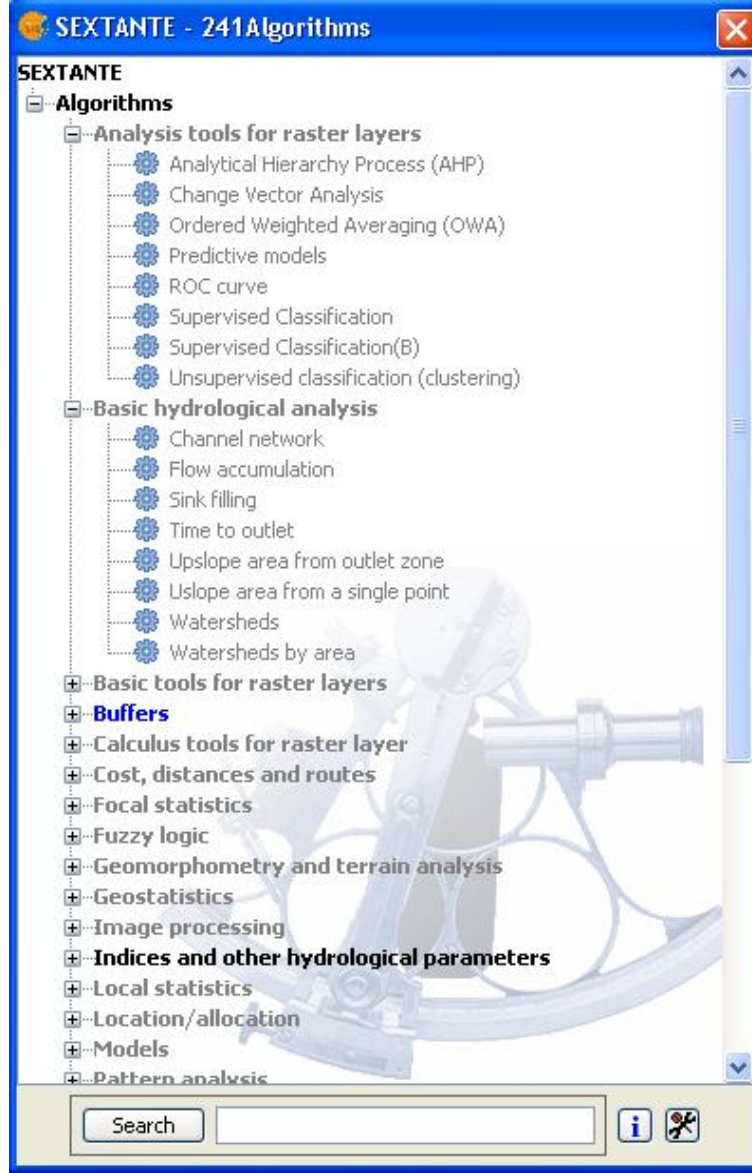
1.4.6. SEXTANTE Projesi

Sextante, İspanya’daki Extremadura yerel idaresi tarafından geliştirilmektedir. Sextante projesi, coğrafi rutinler geliştirmek, bu rutinleri kolayca uygulamak ve kullanmak için bir platform yaratmayı amaçlamaktadır.

Sextante projesi, ilk olarak 2004 yılında, yerel yönetimlerdeki ormancılardan CBS alanındaki ihtiyaçlarını karşılamak üzere geliştirilmeye başlanmıştır. Sextante’nin ilk sürümü, 190 analiz modülüyle, konumsal analizlere ağırlık veren SAGA CBS yazılımı üzerine kurulmuştur. Zamanla CBS alanında Web Servislerine bağlanabilme gibi, SAGA CBS yazılımına göre daha üstün yönlerle gelişen gvSIG CBS yazılımına entegre edilmiştir. (Olaya, 2008)

Günümüzde Sextante; sadece gvSIG için bir genişletme değildir. gvSIG için geliştirilmiş 220 den fazla coğrafi analiz rutini içeren, bağımsız bir açık kaynak kodlu Java kütüphanesidir (Olaya, 2008). Sextante’nin GeoTools’ tan farkı daha özel amaca yönelik rutinler sağlamasıdır.

Sextante’ de ki rutinler kullanım amaçlarına göre yirmi dokuz başlık altında sınıflandırılmıştır. Sextante’de ki bu sınıflar ve içerdikleri rutinlerine örnek olarak; raster katmanlar için analiz araçları (denetimli sınıflandırma, denetimsiz sınıflandırma), temel hidrolojik analizler (basit hidrolojik model), poligon katmanlar için araçlar (polygons to polylines, count points in polygon), vektör katmanlar için araçlar (tampon bölge, fark) örnek gösterilebilir (URL-14, 2009).



Şekil 11. Sextante araç kutusu (toolbox)

Sextante entegre edildiği yazılımlarda, rutinleri yürütmek için kullanıcı arayüzü sağlamaktadır. Şekil 11’de gvSIG programına entegre edilen Sextante’nin araç kutusu (toolbox) gösterilmiştir.

Sextante, sadece gvSIG’e değil diğer CBS uygulamalarına da kolayca entegre edilebilmektedir.

Sextante tek başına verilerle herhangi bir işlem yapamaz, sadece rutinler ve temel sınıflardan oluşmaktadır. Sextante, verilerle işlem yapmak için “veri bağlayıcı” lara (data binding) ihtiyaç duymaktadır. “Veri bağlayıcılar”, rutinleri veri kaynaklarına

bağlamaktadır. Sextante GeoTools 2.4 ve GeoTools 2.5 kütüphanesi için “veri bağlayıcı” geliştirmiştir. Sextante rutinleri bu sayede herhangi bir CBS yazılımına entegre edilmeden, Sextante kullanıcı ara yüzü vasıtasıyla kullanılabilir (Olaya, 2009).

Sextante’yi entegre eden yazılımlar; gvSIG, 52°North WPS ve Sextante’yi entegre etmek için çalışmalar yapan yazılımlar; OrbisGIS, uDig’dir (URL- 15, 2009).

Sextante GPL lisansı altında dağıtılmaktadır.

1.4.7. GRASS (Graphics Resources Analysis Support System)

GRASS ilk olarak, U.S. Army Construction Engineering Research Laboratories tarafından (USA-CERL, 1982–1995) çevresel planlama ve arazi yönetimi için bir araç olarak UNIX ortamında geliştirilmeye başlanmıştır. Geliştirilme amaçlarından biri de o zamanki ticari CBS yazılımlarına bir alternatif yaratmak ve CBS yazılımlarının ücretlerinin azalmasını sağlamaktır. Fakat ESRI firmasının ArcView 3.x serisini çıkartmasıyla birlikte bu yazılımın kullanım kolaylığı nedeniyle geliştirilmesine son verilmiştir. GRASS 1997 yılında ilk ÖAK yazılım olarak tekrar geliştirilmeye başlanmıştır (Westervelt, 2004).

2008 yılından itibaren, OsGeo tarafından desteklenmekte ve GRASS veritabanı (web sitesi, mail listeleri, hata izleyicileri) OsGeo’da bulunmaktadır (URL–16, 2009).

GRASS, veri yönetimi, görüntü işleme, grafik üretme, konumsal modelleme ve birçok veri tipinin görüntülenmesi gibi işlemleri gerçekleştirebilmektedir. GRASS, harita ve görüntülerin ekran üzerinde ve kağıt üzerinde oluşturulması, raster, vektör ve vektör nokta veri listelerinin işlenmesi, multispektral görüntü işleme, konumsal verinin oluşturulması, yönetimi, saklanması ve analizi için 350’den fazla program ve araç sağlamaktadır.

GRASS birçok işletim sistemiyle uyumlu çalışmaktadır. Bu işletim sistemleri; GNU/Linux (Intel, PowerPC, Sun vb.), Solaris (SPARC, i86), SGI IRIX, HP UX, Mac OS X (Darwin), IBM AIX, BSD-Unix çeşitleri, FreeBSD, CRAY Unicos, iPAQ/Linux ve diğer UNIX uyumlu platformlar (32/64bit), MS-Windows native veya Cygnus.

GRASS, GDAL\OGR kütüphanelerini entegre etmektedir. GDAL kütüphanesi, raster veri formatlarını dönüştürmeyi, OGR ise vektör veri formatlarını okumayı sağlayan açık kaynak kodlu kütüphanelerdir. GRASS, birçok raster ve vektör veri formatını desteklemektedir. GRASS’ın desteklediği formatların bir kısmı şu şekildedir;

- Raster formatlar: ASCII, ARC/GRID, E00, GIF, GMT, TIF, PNG, ERDAS LAN, Vis5D, SURFER (.grd)
- Görüntü formatları: CEOS (SAR, SRTM, LANDSAT7 etc.), ERDAS LAN, HDF, LANDSAT TM/MSS, NHAP hava fotoğrafları, SAR, SPOT.
- Topolojik vektör (2D-3D) formatlar: ASCII, ARC/INFO ungenerate, ARC/INFO E00, ArcView SHAPE (topoloji düzeltmesiyle), BIL, DLG (U.S.), DXF, DXF3D, GMT, GPS-ASCII, USGS-DEM, IDRISI, MOSS, MapInfo MIF, TIGER, VRML'dir.
- Sites (vektör nokta veri listeleri): XYZ ASCII, CSV, dBase.

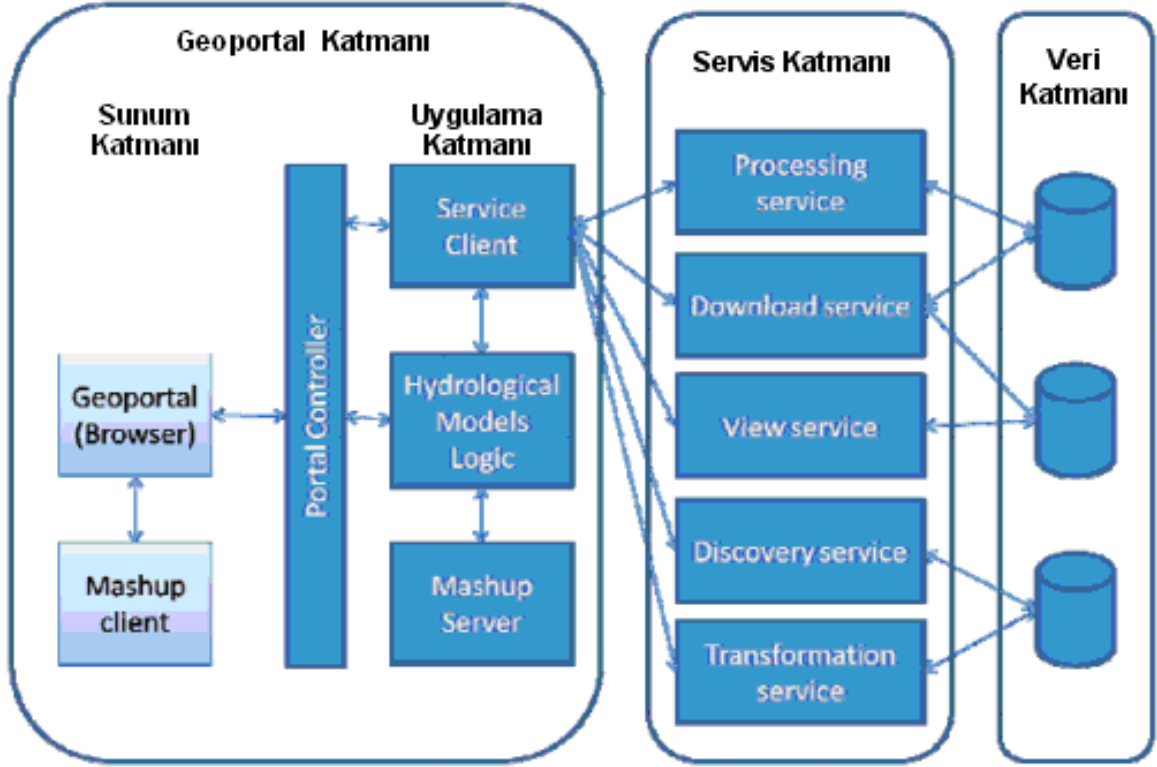
Birçok özellik içermesi ve kendini sürekli güncelleyen bir yazılım olması nedeniyle GRASS çok geniş bir kullanıcı topluluğu tarafından kullanılmaktadır. Günümüzde, akademik, ticari çevre ve kamu kurumları (NASA, NOAA, USDA, DLR, CSIRO, the National Park Service, the U.S. Census Bureau, USGS) tarafından kullanılmakta ve GPL lisansı altında dağıtılmaktadır (URL-16, 2009).

1.4.8. AWARE (A tool for monitoring and forecasting Available Water Resource in mountain environments) Projesi

Bu alanda yapılan önemli çalışmalardan biri (Díaz, vd., 2007)'nin yapmış oldukları çalışmadır. Bu çalışma kapsamında; çok büyük hacimli verilerle çalışmayı gerektiren hidroloji alanında, ihtiyaç duyulan konumsal analiz ve işlemlerin ticari masaüstü coğrafi bilgi sistemleri uygulamaları ile yapılması yerine, açık kaynak kodlu yazılımlar ve servis yönelimli mimari kullanılarak geliştirilen, temel konumsal veri altyapı işlevleri üzerine inşa edilen dağıtık geoprocessing servis modeli önerilmiştir.

Bu bağlamda hidrolojik modelleme uzmanları ve bilim adamlarına yönelik kullanılması kolay ve enteraktif bir ara yüz sağlamak için, konumsal işlem servisleri ve web harita görüntüleyicilerini bütünleştiren, hidrolojik uygulamalara yönelik web tabanlı bir coğrafi portal (geoportal) geliştirilmiştir (Díaz, vd., 2008).

AWARE uygulama mimarisi, standart ara yüzler ve yeniden kullanılabilir bileşenlere dayalı açık ve birlikte işleyen (interoperable) bir mimari kurmak için INSPIRE teknik mimarisini takip etmiştir. Ana amacı kullanıcı ihtiyacını karşılayacak servisleri belirlemek ve bu servislerin yeniden kullanılabilirliğine ve kompozisyonuna olanak sağlamaktır.



Şekil 12. AWARE mimarisine genel bakış (Díaz, vd., 2008).

Şekil 12’de AWARE projesinin genel mimarisi gösterilmiştir. Geoportall katmanı, iki katmandan oluşmuştur. Sunum katmanı kullanıcı ara yüzü için gerekli yazılımları içerir ve kullanıcının geoportall ile etkileşimini sağlar. Uygulama katmanı, servis yürütme, business logic ve servis zincirleme kontrolü için sabit bileşenlerin tanımlanmasına ve uygulanmasına izin verir. Servis katmanı, INSPIRE servis tiplerine göre gruplandırılmış servislerden oluşmaktadır. Bu servisler; discovery, view, download, transformation ve processing servisleridir. Veri katmanında, veri setleri ve metaveriler bulunmaktadır.

AWARE projesinde hidrolojik modelleri koşturmak için gerekli atomik işlevler belirlenerek, ortak atomik işlevlerden benzer olanlar aynı modül içerisinde toplanarak gruplandırılmıştır (Díaz, vd., 2008). Her bir modül web servisi olarak tasarlanarak, WPS olarak uygulanmıştır. AWARE uygulamasındaki WPS servisleri, Tablo 4’de gösterilmiştir.

Tablo 4. AWARE WPS (Díaz, vd., 2008)

Web Process Servisleri	Servis Türü	Servis İşlemleri
DataConversion W PS	Transformation Service	CoordTransformer, Shp2GML
TopologyWPSS	Processing Service	Area, Intersection, Buffer, MaxExtent, SnowPercentage, GetFeatureByAttribute, Thiessen
SextanteWPS	Processing Service	CoordinateElevation, StationsElevation, ElevationCurves, ElevationZones, HypsometricElevation, Reclassify, Vectorize
SRMIDLWPS	Processing Service	SRMIDL
ChartWPS	View Service	DepletionCurvesPlot, DischargePlot, HbvRunoffPlot, HbvSWEPlot, SensorDataChart

Uygulama Örneği: Sextante WPS

AWARE uygulamasında; SEXTANTE, GeoTools ve JFreeChart vb. ÖAK yazılım kütüphanelerinin fonksiyonları, OGC WPS uygulaması olan 52N WPS çatısına entegre edilmiştir. Uygulama kapsamında, SEXTANTE fonksiyonları sarmalanarak, SEXTANTE WPS servisi oluşturulmuştur (Diaz, 2008).

Örnek bir WPS iş akışını ele alacak olursak, havza üzerindeki bir alana ait yükseklik sınıflarını belirleyen bir işlem (ElevationZones) için izlenecek adımlar aşağıdaki gibidir;

- Yükseklik bölgeleri hesaplanacak alana ait SYM (Sayısal Yükseklik Modeli) dosyası, projeksiyonu ve yükseklik aralığı girdi olarak verilir. Bunun üzerine SEXTANTE WPS'teki "ElevationZones" işlemi yürütülerek yükseklik bölgeleri hesaplanır.
- Yükseklik bölgeleri hesaplanan alanın tüm havza ile kesişimi hesaplanır. Bu işlem, Topology WPS servisindeki "intersect" işlemi tarafından yapılır.
- Her bir yükseklik bölgesi, kategorisine göre yükseklik bölgeleri dosyasından çıkartılır (extracted).
- Topology WPS' teki "getFeatureById" işlemi ile ana dosyadaki yükseklik alanı çıkartılır.
- 3. işlemin sonucunda elde edilen yükseklik bölgeleri, GML formatından KML (Keyhole Markup Language) formatına dönüştürülerek sunum katmanında harita görüntüleyici tarafından gösterilir.

- 4. işlemin sonucunda elde edilen yükseklik alanı ise koordinat dönüşümü yapıldıktan sonra sunum katmanında GML formatından KML formatına dönüştürülerek harita görüntüleyici tarafından gösterilir (Diaz, 2008).

1.4.9. GeOnAs (GeoBrain Online Analysis System)

GeoBrain, açık, birlikte işleyen, standartlara –uyumlu, çok katmanlı, web tabanlı coğrafi bilgi sistemi servisleridir. Sistemin amacı, NASA ESE'nin (Earth Science Enterprise) sağladığı EOS (Earth Observing System) verilerine kullanıcıların web üzerinden kolayca erişmesini ve analizler yapmasını sağlamaktır (Di, 2004).

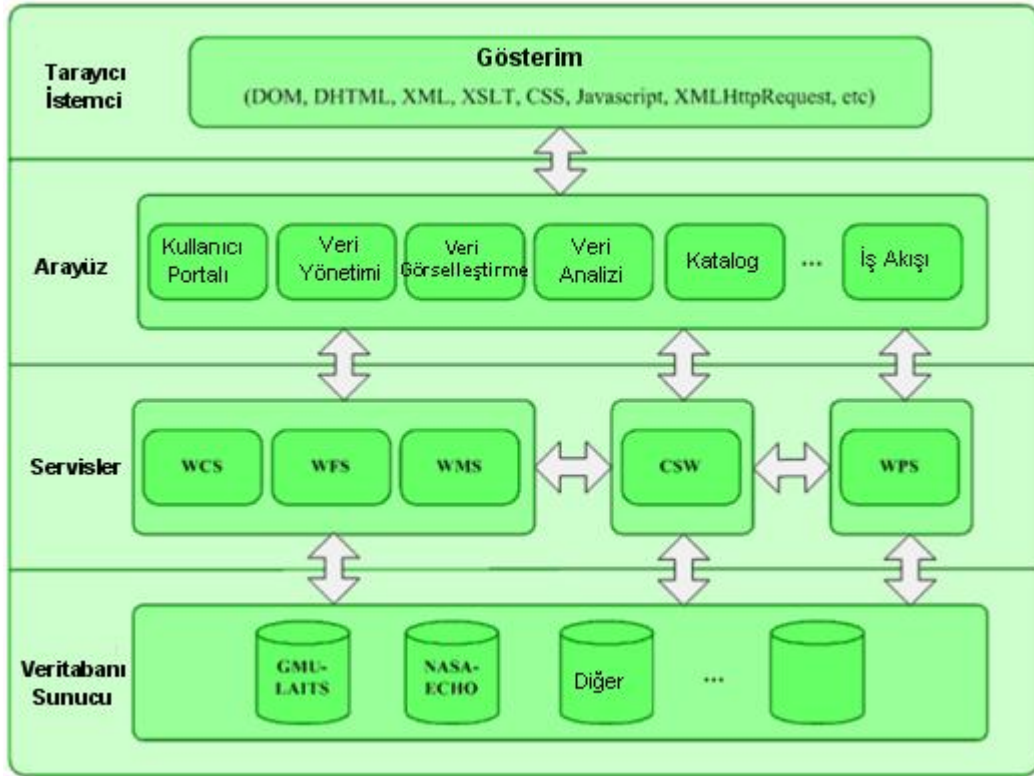
GeoBrain sistemi, web üzerinden veri bulma ve erişmede OGC standartlarını, web servislerinde ise W3C standartlarını kullanmaktadır.

Şekil 13'te GeOnAs mimarisi gösterilmiştir. GeOnAs mimarisinde, kullanıcı, Firefox Mozilla veya Internet Explorer gibi bir web tarayıcısıdır. Kullanıcıların GeOnAs'ı kullanmak için yeni bir program kurmaya ihtiyaçları yoktur.

Interface katmanı, kullanıcılara verilerin yönetimi, görüntülenmesi ve konumsal analizlerin yapılması için gerekli ara yüzleri sağlamaktadır.

Servis katmanı; OGC uyumlu web servislerinden oluşmuştur. CSISS (Center for Spatial Information Science and System), George Mason Üniversitesi' nin Web Coverage servisinin yanı sıra, WMS, WFS, WCS (Web Coverage Service), WPS servislerini uygulamıştır.

Veritabanı servisi, konumsal veri ve bilginin saklandığı veri havuzundan oluşmaktadır. GMU-LAITS ve NASA- ECHO veritabanlarına erişilebilmektedir. GMU-LAITS, CSISS tarafından kurulmuştur ve LANDSAT (ETM, TM, MSS), SRTM, ASTER(L1B in Geo- TIFF, L1B in HDF, L2 in HDF), BlueMarble vb. 12 terrabyte raster veri sağlamaktadır. Bu veri setleri, GeOnAs tarafından çağrılabilir, dönüştürülebilir ve çıkartılabilir (extracted).



Şekil 13. GeOnAs genel mimarisi (Han, vd., 2008)

GeOnAs, modüllerden oluşmaktadır. Bu modüller; proje yönetimi, veri işleme, harita operasyonları ve görüntüleme, vektör ve raster veri analizi, web servisi yürütme ve yardımcı işlemlerdir.

Proje yönetimi; projelerin saklanması, oluşturulması gibi temel özellikleri sağlamaktadır. Proje dosyaları WMC veya KML (Keyhole Markup Language) formatında saklanmaktadır.

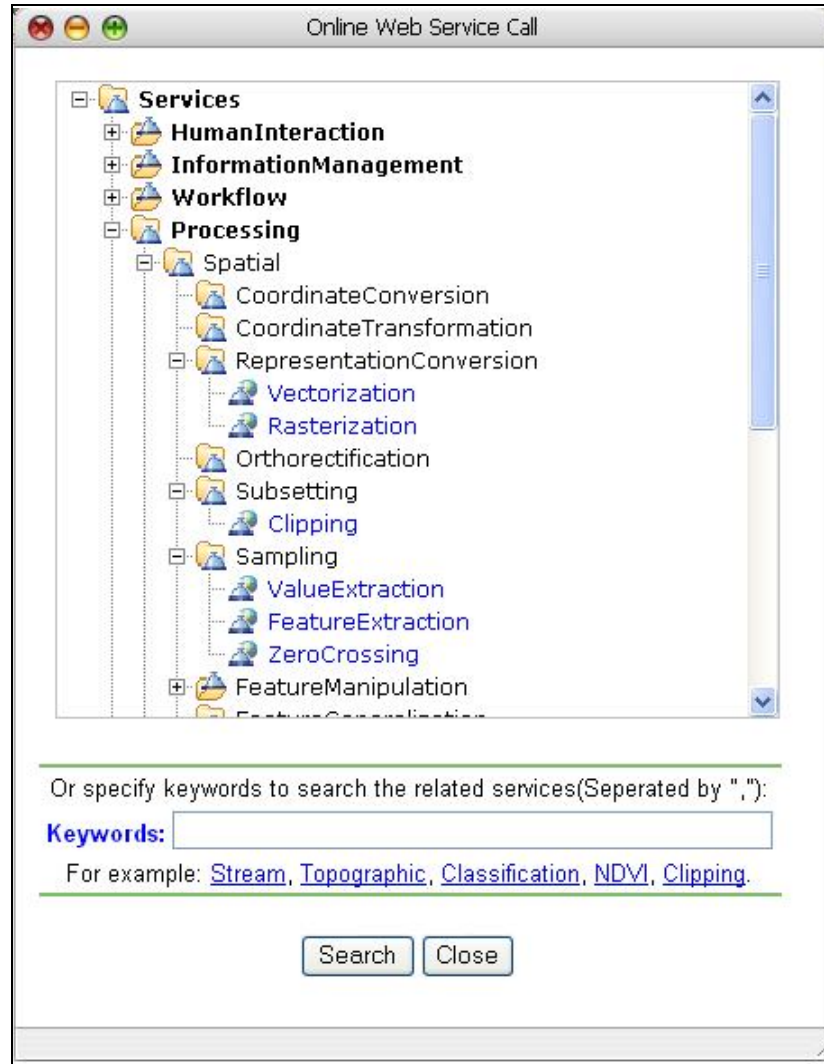
Veri işleme; verinin sorgulanması, işlenmesi, seçilmesi, eklenmesi veya çıkarılması vb. işlemlerin yapılmasını sağlamaktadır.

Harita operasyonu ve görüntülemek; yakınlaştır-uzaklaştır, tam ekran gibi özellikleri sağlamaktadır.

Vektör ve raster veri analizleri; GeOnAs da 20'den fazla WPS içerisinde ve 50'den fazla işlem uygulanmıştır. Vektör veri analizleri, detay sorgulama, detay çıkartma, en kısa yol servilerinden oluşmaktadır. Raster veri analizleri; NDVI, görüntü işlemleri, renk dönüşümü (RGB2HIS, HIS2RGB), RGB kompozisyonu (true color, false color), kesme, image patch, image mosaic, sınıflandırma (denetimli ve denetimsiz) servislerinden oluşmaktadır.

GeOnAs raster ve vektör analiz servisleri, açık kaynak kodlu bir CBS yazılımı olan GRASS fonksiyonları sarmalanarak geliştirilmiştir. Servislerin detaylı tanımlamaları, içerdikleri operasyonlar, girdi ve çıktı parametreleri, servis location, WSDL dokümanları, <http://geobrain.laits.gmu.edu:81/grassweb/manuals/index.html> adresinde bulunmaktadır. GeOnAs, kullanıcılara kendi WPS servislerini entegre etme olanağı da sağlamaktadır (Di, vd., 2007).

GeOnAs kullanıcı ara yüzüne <http://geobrain.laits.gmu.edu:81/OnAS/> adresinden erişilebilmektedir. Şekil 14' te GeOnAs online web servisleri kullanıcı ara yüzü gösterilmiştir.



Şekil 14. GeOnAs online web servisleri kullanıcı ara yüzü

Ayrıca GeOnAs kullanıcılara, geliştirdiği GRASS tabanlı konumsal web servislerini kendi web servisleri olarak kullanabilmeleri imkânı da sunmaktadır. Tablo 5’te, geliştirilen GRASS tabanlı vektör harita işleme web servisleri ve açıklamaları gösterilmiştir.

Tablo 5. Grass tabanlı konumsal vektör harita işleme servisleri (URL- 17, 2009)

Web Servisi Adı	Web Servisi Açıklaması
Grass_Vector_AttributeColumn Service	Servis, vektör bir haritanın öznelik bilgilerini yazar.
Grass_Vector_Buffer Service	Coğrafi objelerin etrafında tampon bölge oluşturur.
Grass_Vector_ClippingService	Servis, iki vektör haritanın kesişim, birleşim, overlay ve cutout işlemlerini yapar.
Grass_Vector_CompositeService	Bir grup vektör katmanının tek bir vektör katmanı olarak birleştirilmesini(composite) sağlar.
Grass_Vector_DataConversionService	Servis, GML ve shapefile formatları arasında dönüşüm yapmayı sağlar.
Grass_Vector_FeatureExtractionService	Vektör bir haritadan vektör detayları çıkartmak (extract) için kullanılır.
Grass_Vector_RasterizationService	Servis, vektör haritayı raster haritaya dönüştürür.
Grass_Vector_ShortestPathService	Servis, en kısa yol analizi yapar.
Grass_Vector_TopologyManagementService	Servis, vektör haritanın topolojisini oluşturur, değiştirir ve düzeltir (repair).
Grass_Vector_ValueExtractionService	Servis, Raster bir haritadan vektör bir haritadaki noktaların konumuna göre raster değerleri çıkarır.

2. YAPILAN ÇALIŞMALAR

2.1. Örnek Uygulamanın ÖAK Yazılımlarla Gerçekleştirilmesi

Bu bölümde örnek uygulama olarak; kadastral işlemlerden birisi olan aplikasyon krokisinin hazırlanması işleminin ÖAK yazılımlar kullanılarak konumsal web servislerine yaptırılması amaçlanmıştır. Uygulamada ÖAK yazılımların tercih edilmesinin temel sebebi; belirlenen web servislerinin gerçekleştireceği işlemler için gereken rutinleri ÖAK yazılımlardan hazır olarak alıp kullanarak zamandan tasarruf etmek, var olan bir rutini tekrar yazmamaktır.

2.1.1. Örnek Uygulama İşlem Adımları

Örnek uygulama sırasında gerçekleştirilen işlem adımları aşağıda sırasıyla belirtilmiştir, izleyen kısımda da ayrıntılı olarak anlatılmıştır.

1. Uygulama için gerekli olan Web servislerinin belirlenmesi.
2. Web servislerinin geliştirilmesinde kullanılacak olan ÖAK yazılımının belirlenmesi.
3. Web Servislerinin geliştirilmesi.
4. Web Servislerinin Web uygulama sunucusuna kurulması (deploy) ve koşturulması.

2.1.2. Örnek Uygulama İşlem Adımlarının Ayrıntılı Açıklaması

2.1.2.1. Uygulama İçin Gerekli Olan Web Servislerinin Belirlenmesi

Uygulama kapsamında, Aplikasyon Krokisi hazırlamak için gerekli web servisleri belirlenmiştir. Belirlenen web servisleri, konumsal web servisleri Tablo 6 ve konumsal olmayan web servisleri Tablo 7 olmak üzere iki gruba ayrılmıştır. Tablolarda web servislerine ait kısa açıklamalar yapılmıştır. Bu çalışma kapsamında, konumsal web servisleri ele alındığı için bu servislere ait girdi ve çıktı parametreleri tabloda belirtilmiştir.

İlgili parselin ve gerektiğinde üzerindeki yapı sınırlarının, orijinal kadastral ölçülerinden elde edilen değerler kullanılarak ve poligon noktalarına dayalı olarak

ölçülmesi ve parsel köşe noktalarının zeminde işaretlenmesi işlemidir. Tanımlamadan da anlaşılacağı üzere işlem temel iki aşamadan oluşmaktadır. Önce büroda aplikasyon işlemine temel altlık teşkil eden aplikasyon krokisi hazırlanır sonra arazide ölçü işlemleri gerçekleştirilir.

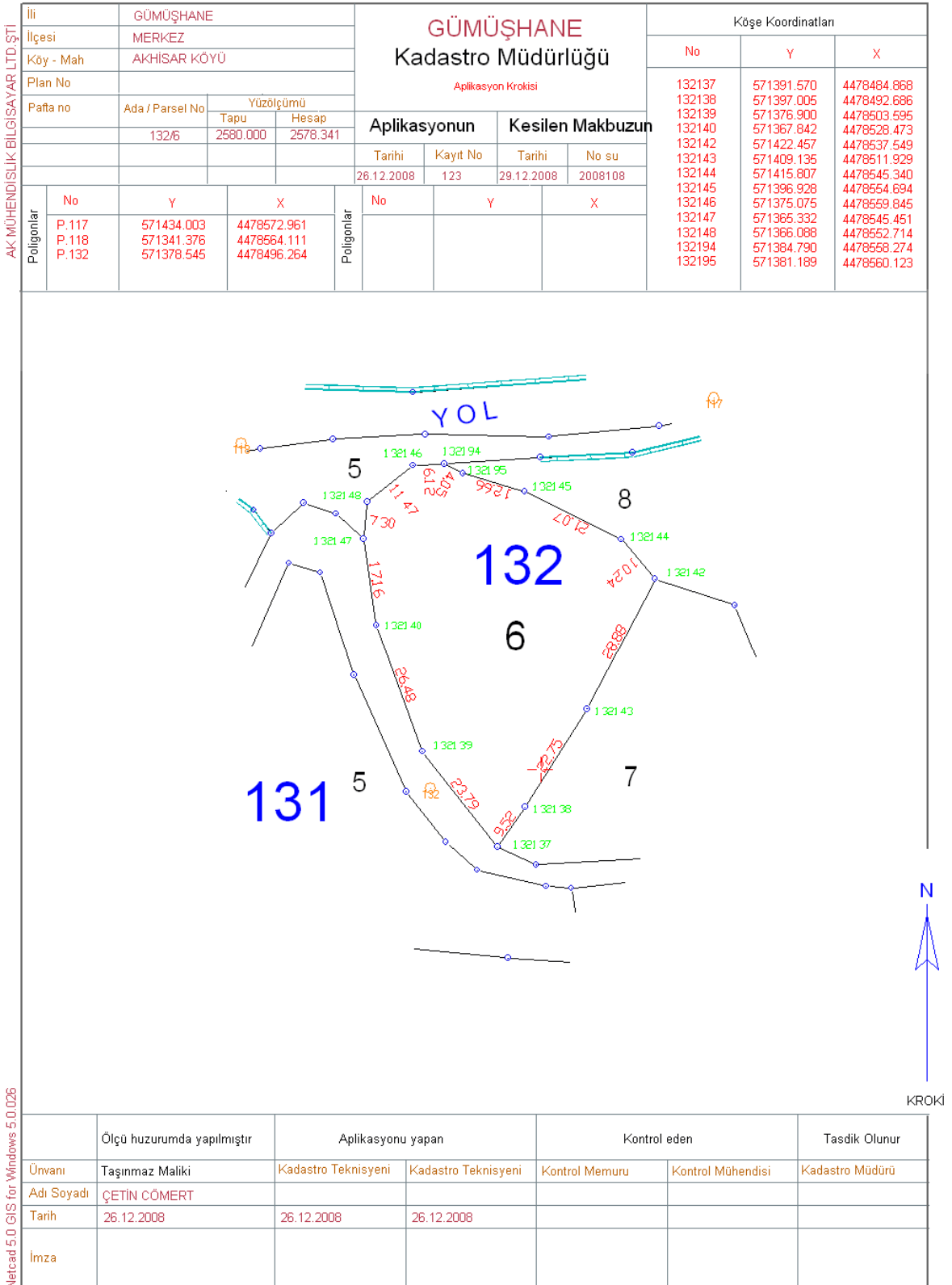
Aplikasyon Krokisi; İlgili parsel sınırlarının ve komşuluk ilişkilerini göstermeye yetecek kadarıyla komşu parsel sınırlarının gösterildiği, parselin ve varsa üzerindeki yapının cephe uzunlukları ile köşe no' larının yazıldığı, komşu parsel no' larının ve cadde sokak isimlerinin de yer aldığı grafik bölüm ile il-ilçe-mahalle adı, pafta-ada-parcel no, parselin tapu ve hesap alanı, parsel köşe noktalarının ve aplikasyonda referans alınacak poligon noktalarının numaralarının ve koordinat değerlerinin, işlem tarihinin, ilgili bilgisinin ve işleme ait kayıt ve makbuz no bilgisinin yazıldığı çizelge kısımdan oluşan, aplikasyon işlemine teknik ve hukuki altlık teşkil eden bir belgedir. Şekil 15'de örnek aplikasyon krokisi gösterilmiştir.

Tablo 6. Konumsal web servisleri

Web Servisleri	Açıklama	Girdi	Çıktı
ParselAlanHesapla	Parselin, hesaplanan alanını döndürür.	Parsel_ID	Parselin Hesaplanan Alanı
ParselGeometriGetir	Parselin, geometrik şeklini (sınırlarını) döndürür.	Parsel_ID	Parsel Geometrisi
BinaGeometriGetir	Parselin, üzerinde yapı varsa bu yapının/yapıların geometrik şeklini (sınırlarını) döndürür.	Parsel_ID	Bina Geometrisi
ParselKöşeKoordinatGetir	Parselin, köşe koordinatlarını döndürür.	Parsel_ID	Parsel Köşe Koordinatları
BinaKöşeKoordinatGetir	Parselin üzerinde yapı varsa bu yapı/yapılara ait köşe koordinatlarını döndürür.	Parsel_ID	Bina Köşe Koordinatları
ParselCepheHesapla	Parselin, cephe uzunluklarını hesaplar.	Parsel_ID	Parsel Cephe Uzunlukları
PoligonKoordinatGetir	Parsele tampon bölge (buffer) uygulayarak buffer alanı içerisinde kalan poligon noktalarının koordinatlarını döndürür.	Parsel_ID	Poligon Noktaları ve Koordinatları
KomşuParselSınırÇiz	Parsele tampon bölge (buffer) uygulayarak, komşu parsellerinin buffer alanı içerisinde kalan sınırlarını döndürür.	Parsel_ID	Komşu Parsel Sınırları

Tablo 7. Konusal olmayan web servisleri

Web Servisleri	Açıklama
İlAdıGetir	Parselin bulunduğu ilin adını döndürür.
İlçeAdıGetir	Parselin bulunduğu ilçenin adını döndürür.
MahalleAdıGetir	Parselin bulunduğu mahallenin adını döndürür.
PaftaNoGetir	Parselin bulunduğu paftanın no' sunu döndürür.
AdaNoGetir	Parselin, ada no' sunu döndürür.
ParselNoGetir	Parselin parsel no' sunu döndürür.
TapuAlanGetir	Parselin, tapu alanını döndürür.
AplikasyonNoYaz	Parselin, aplikasyon no' sunu döndürür.
AplikasyonTarihiYaz	Parselin, aplikasyon işlemi başvuru tarihini döndürür.
MakbuzTarihiYaz	Parsele ait aplikasyon işlemi için hesaplanan döner sermaye ücretinin ödendiği makbuz tarihini döndürür.
MakbuzNoYaz	Parsele ait aplikasyon işlemi için hesaplanan döner sermaye ücretinin ödendiği makbuz tarihini döndürür.
PoligonRöperGetir	Poligona ait röper krokisini döndürür.



Şekil 15. Örnek aplikasyon krokisi

2.1.2.2. Konumsal Web Servislerinin Geliştirilmesinde Kullanılan ÖAK Yazılımın Belirlenmesi

Bu aşamada, konumsal web servislerinin gerçekleştirilmesi için, konumsal ÖAK yazılımlar araştırılmıştır. Buradaki amacımız, uygulama kapsamında belirlediğimiz konumsal web servislerini geliştirmek için, ihtiyacımız olan yazılım veya yazılımların belirlenmesidir.

2.1.2.2.1. Ön Seçim

Uygulamaya yönelik mevcut yazılımları belirlemek ve bu yazılımlar hakkında fikir edinmek amacıyla, ilk aşamada konumsal alandaki ÖAK yazılımlar genel olarak araştırılmıştır.

Araştırma kapsamında; konumsal ÖAK yazılımlar ile ilgili literatür taraması yapılarak, çalışma konusuna paralel projeler (Aware, GeOnAs) ve ÖAK yazılımları yayımlayan web siteleri araştırılmıştır.

Bu alandaki web sitelerinin birçoğu, ÖAK yazılımları destekleyen vakıflar tarafından, destekte buldukları ve destekledikleri yazılımla ilgili olan diğer yazılımları da duyurmak amacıyla kurulmuştur. Web sitelerinden sadece konumsal ÖAK yazılımları içeren (OsGeo.org, MapTools.org vs.) sitelerin yanı sıra, genel ÖAK yazılımları içeren web siteleri de araştırma kapsamında incelenmiştir (sourceforge.net, fsf.org).

Araştırma kapsamında incelenen web siteleri ve yapılan literatür araştırmalarında ÖAK yazılımlarının çoğunlukla farklı ölçütlere göre gruplandırılarak anlatıldıkları belirlenmiştir. Bu nedenle araştırılan konumsal ÖAK yazılımlar, gruplandırma ölçütleri ile birlikte yazının devamında anlatılmıştır.

2.1.2.2.1.1. Literatür Araştırması

(Hall ve Leahy, 2008)'de, konumsal alandaki ÖAK yazılımları kullanıcılara tanıtmak için OsGeo (Open source geospatial community) vakfının desteklediği projelerden bir kısmı ile konumsal ÖAK yazılımlar ile ilgili diğer projelere yer verilmiştir. Çalışmada yer alan yazılımlar, ilgili projelerde aktif rol alan kişiler tarafından anlatılmıştır. Anlatılan yazılımların belirlenmesinde herhangi bir ölçüt kullanılmamıştır.

Çalışma kapsamında anlatılan konumsal ÖAK yazılımlar: MapServer, The Geospatial Data Abstraction Library (GDAL), MapGuide Open Source, GeoTools, GRASS CBS, GeoVISTA Studio, MapChat Project, TerraLib'dir.

“Genel Bilgiler” bölümünde de anlatılan örnek projelerden, AWARE projesinde konumsal web servislerinin geliştirilmesi için; GeoTools, Sextante, JTS, JAI (Java Advanced Image API) ÖAK kütüphaneleri ve uygulama programlama ara yüzleri kullanılmıştır. Bu nedenle araştırma kapsamında projede kullanılan bu yazılımlar incelenmiştir.

GeOnAs projesinde geliştirilen GRASS tabanlı konumsal vektör web servisleri de yine bu araştırma kapsamında incelenmiştir.

(Ramsey, 2007) çalışmasında, ÖAK yazılımlarını geliştirilen programlama dili ve bir uygulama veya kütüphane olup olmamasına göre sınıflandırmıştır Tablo 8 Ayrıca konumsal ÖAK web yazılımlarını da türlerine göre sınıflandırmıştır. Bu sınıflandırma Tablo 9'da gösterilmiştir.

Tablo 8. (Ramsey,2007)'in ÖAK yazılımlarını uyguladıkları programlama dillerine ve kütüphane veya yazılım olmalarına göre yaptığı sınıflandırma

Kategoriler	C Projeleri	Java Projeleri	. Net Projeleri
Kütüphaneler	GDAL/OGR, Proj4, GEOS, Mapnik, FDO,	JTS, GeoTools	NTS, ProjNET, SharpMap,
Uygulamalar	MapGuide, UMN MapServer, GRASS, QGIS, OSSIM, TerraLib, Gmt, PostGIS	GeoServer, Deegree, JUMP/ OpenJUMP/Cosmo, gvSIG, OpenMap, UDig	WorldWind, MapWindow

Tablo 9. (Ramsey,2007)'in konumsal ÖAK web yazılımlarını türlerine göre yaptığı sınıflandırma

Kategoriler	Web Projeleri
Araç kutuları (Toolkits)	MapBuilder,Ka-Map!, OpenLayers
Çatılar (Frameworks)	MapBender, CartoWeb
Sunucular (Servers)	TileCache, FeatureServer

Bu kapsamda ki araştırmalardan biri de (Steiniger, 2008) tarafından mevcut ÖAK masaüstü CBS yazılımlarının değerlendirilmesi konulu çalışmasında yapılmıştır.

Çalışmada “anlatmaya değer” yazılımları seçmek için bazı ölçütler belirlenmiştir. Bu alanda dikkate alınan ölçütler Tablo 10’da gösterilmiştir. Ölçütler sonucunda çalışma kapsamında anlatılan ÖAK masaüstü yazılımlar; GRASS, QGIS, uDig, gvSIG, Kosmo, SAGA, ILWIS, MapWindow, Jump/OpenJUMP yazılımlarıdır.

Tablo 10. (Steiniger, 2008)’in ÖAK yazılımları seçme ölçütleri

Ölçütler
Kullanım Alanı
Kullanıcı Düzeyi
İşletim Sistemi
Programlama Dili
Kim Tarafından Geliştirildiği
Yazılım Lisansı

2.1.2.2.1.2. Genel ÖAK Yazılımları İçeren Web Siteleri

Özgür yazılım kavramını da literatüre kazandırmış olan FSF vakfının web sitesinde desteklediği her projeye ilgili; kısa özet bilgi, yazılımın sürümleri, kaynak arşivinin adresi, kullanıcılar için; yazılımın web adresi, geliştiriciler için; hata izleme kaynakları, e-posta listelerinin adresleri ve geliştirmek üzere kaynak koduna erişebilecekleri adresler hakkında bilgiler verilmiştir.

FSF vakfı, birçok alanla ilgili ÖAK yazılım yayınlamaktadır. Yayınladığı yazılımları kullanıldığı alana göre kategorilere ayırarak sunan, <http://directory.fsf.org/>, bilimsel alandaki ÖAK yazılımları “bilim” kategorisi altında yayınlamaktadır. Listelenen bilim kategorileri Tablo 12’de gösterilmiştir.

Bu adreste listelenen yazılımlar Tablo 11’de belirtilen kategorilere göre gruplandırılmıştır.

Tablo 11. fsf.org sitesindeki yazılımların kategorileri

a)	
Kategoriler	
Sesle İlgili (Audio)	
İş ve Verimlilik	
Veritabanı	
Eğitim	
E-mail	
Oyunlar	
Grafik	
Hobi	
Ara yüz	
İnternet Uygulamaları	
İletişim	
Yerelleştirme	
Matematik	
Baskı (printing)	
Bilim	
Güvenlik	
Yazılım Geliştirme	
Yazılım Kütüphaneleri	
Sistem Yönetimi	
Metin oluşturma ve Düzenleme	
Video	
Web yazma (Web authoring)	
	b)
	Bilim Kategorileri
	Astronomi
	Yapay Zekâ
	Fizik
	Jeoloji
	Kimya
	Biyoloji
	Coğrafya
	Bilimsel Görselleştirme
	Karışık

Konumsal alandaki yazılımlar “coğrafya” kategorisi altında yayınlamaktadır. Coğrafya kategorisi altında yayınlanan ÖAK yazılımlar: DGPS, Earth3D, GDAL, GEOS, GMT, GPSBabel, GPStk, GRASS, GeoToad, GpsDrive, Harmonics, Height Map

Generator, MapServer, PostGIS, QGIS, RadMap, Thuban, XTide, Xrmap, gama, tkgeomap yazılımlarından oluşmaktadır.

Açık kaynak kodlu projelerin geliştirilmesi için servisler sağlayan <http://sourceforge.net/> birçok alanda projeye ev sahipliği yapmaktadır. Şubat 2009 itibariyle sitede birçok alanla ilgili 230.000'den fazla açık kaynak kodlu proje kayıtlı bulunmaktadır.

Web sitesinde açık kaynak kodlu yazılımlar, genel uygulama alanlarına göre kategorilere ayrılmıştır. Bu kategoriler Tablo 12'de belirtilmiştir. Her kategori yine kendi içinde birçok kategoriye ayrılmıştır. Ayrıca yazılımları anahtar kelime girerek ve işletim sistemini seçerek arama yapma olanağı sağlar. Bir diğer gruplandırma yöntemi de yazılımları popüler ve aktif olma durumlarına göre listelemektedir. Sayfada güncel olarak hangi alanda olduğu fark etmeksizin son yedi gün içerisindeki en aktif ve en popüler yazılımların listesi de verilmektedir.

Tablo 12. sourceforge.net sitesindeki yazılım kategorileri

Kategoriler
Kümeleme (Clustering)
İletişim
Veritabanı
Masaüstü Ortamı
Yazılım Geliştirme
Eğitim
Girişim
Finans
Oyun/Eğlence
Donanım
Ağ Oluşturma
Güvenlik
Depolama (Storage)
Sistem Yönetimi
VoIp

Sitede konumsal alanla ilgili yazılımları bulmak için anahtar kelime kullanarak arama yapılmıştır. GIS (Geographic Information System) anahtar kelimesine göre arama yapıldığında, bu alanla ilgili yazılımlar listelenmektedir (Şekil 16). Genel ÖAK yazılımları yayınlayan bir site olduğu için, konumsal alanda ki yazılımları herhangi bir ölçüte göre gruplandırmamıştır. Kullanıcılara, listelediği yazılımlarla ilgili dört başlık altında bilgi vermektedir; özet, dosyalar, destek, geliştirme.

Özet kısmında yazılımla ilgili genel açıklamalar, dosyalar kısmında yayınlanan bütün sürümlerin listesi, destek kısmında projeye ilgili yardım alınabilecek mail listeleri, projenin web adresi ile ilgili bilgiler yer almaktadır. Geliştirme kısmında ise; yazılımın programlama dili, geliştiricilerin listesi ve programa yaptıkları katkılar güncel olarak listelenmektedir.

SourceForge.net - Search

Welcome, Guest! [Log In](#) | [Create Account](#)

[Find Software](#) | [Develop](#) | [Create Project](#) | [Community](#) | [Site Support](#) | [About](#)

Search: enter keyword [Search](#)

Sponsored Links

- [ENTEC Reoloser](#)
Reoloser, LBS, GIS, DC Switchgear DNP3.0, Microprocessor Controller
[www.entecene.co.kr](#)
- [Gis Suppliers](#)
Thousands of Prequalified Suppliers Trade Leads, Products & Companies
[Alibaba.com](#)
- [Nemokama GIS programı](#)
MapWindow - nemokama GIS programine [ranga, Parsislişkikte Ir naudokite
[www.infoera.it](#)
- [LG Arena KM900](#)
Üç boyutlu arayüzüyle 3G özellikli Cep telefonu KM9000 deneyin
[tr.lge.com/cep-telefonu](#)

Search results in projects found for "GIS" [Search Help](#)

Results 1 - 15 of 248 [Display:Details](#) [Pages](#) [Filters](#) View: [15](#)

Exact matches found: [Gentoo Installation System](#)

Name	Relevance	Activity	Rank	Registered	Latest File	Downloads
SAGA GIS	80.81%	99.96%	117	2004-22-	2008-70-	235,671
SAGA is a free geographic information system (GIS), with a special 'Application Programming Interface' (API) for geographic data processing. This API makes it easy to implement new algorithms. The SAGA API supports grid data, vector data, and tables.						
Topic: Visualization , GIS , Earth Sciences , Simulations , Ecosystem Sciences , Frameworks						
GeoAPI	57.14%	99.64%	949	2002-22-	2008-61-	11,813
The development community in building GIS solutions is sustaining an enormous level of effort. The GeoAPI project aims to reduce duplication and increase interoperability by providing neutral, interface-only APIs derived from OGC/ISO Standards.						
Topic: GIS						
GeoTools, the java GIS toolkit	80.81%	99.57%	1,133	2000-32-	2009-60-	405,541
GeoTools is an open source java GIS toolkit. Used for OGC based projects via GeoAPI interfaces. Includes two great SLD based renderers, raster access and reprojection. Plugins fo Shapefile, ArcGrid, ArcSDE, Postgis, OracleSpatial, MySQL and many more.						
Topic: Viewers , Visualization , WWW/HTTP , GIS , Front-Ends , Editors						

Şekil 16. sourceforge.net sitesindeki “GIS” anahtar kelimesine göre arama sonuçlarını gösteren ara yüz

OSOR (Open Source Observatory and Repository) <http://www.osor.eu/> Avrupa kamu kurumlarında kullanılmak üzere, bilgi, deneyim ve FLOSS (Free/Libre/Open Source) tabanlı kod alışverişi için kurulmuş bir platformdur. OSOR birçok alana ait açık kaynak kodlu projeleri içermektedir.

Yazılımları sınıflandırmak için birçok kategori belirlenmiştir. Kategoriler genel olarak; yazılım lisansı, programlama dili, kullanıcı düzeyi, kullanım alanına yönelik kategorilerden oluşmaktadır.

Projeler hakkında; genel bir açıklama, lisans bilgisi, çalıştığı işletim sistemi, programlama dili vb. bilgileri sağlamaktadır. Projenin kaynak koduna ve varsa projeye ilgili diğer dokümanlara erişmeyi sağlamaktadır.

2.1.2.2.1.3 Konumsal ÖAK Yazılımları İçeren Web Siteleri

OsGeo Vakfı (The Open Source Geospatial Foundation); kaliteli açık kaynak kodlu coğrafi yazılımları hazırlamak ve bunlara destek vermek için 2006 yılında kurulmuştur. Vakfın amacı topluluk destekli projelerin kullanımını ve ortak geliştirilmesini teşvik etmektir. Web sayfası, <http://www.osgeo.org/>, kullanıcılara ve geliştiricilere fikirlerini paylaşmak ve proje gelişimine katkı sağlamak için portal olarak sunulmaktadır. Vakıf, desteklediği projeleri web sayfasında uygulama Tablo 13'deki gibi türlerine göre sınıflandırmıştır.

Tablo 13. Osgeo.org sitesindeki yazılım kategorileri

Kategoriler	Yazılımlar
Web Haritacılığı	Deegree, Mapbender, MapBuilder, MapGuide, MapServer, OpenLayers
Masaüstü Uygulamaları	GRASS CBS, OSSIM, Quantum GIS, gvSIG
Konumsal Kütüphaneler	FDO, GEOS, GDAL/OGR, GeoTools, MetaCRS, PostGIS
Metadata Katalog	GeoNetwork

<http://opensourcegis.org/> web sayfası, açık kaynak ve özgür CBS yazılımlarıyla ilgili projelerin, bir indeksini oluşturmak için hazırlanmıştır. Sitede, yazılımlar hakkında kısa açıklamalar yapılarak yazılımların lisans bilgileri de belirtilmiştir. Bu açıklamalar

yazılımların web sitelerinden alınmıştır. Sitede yayınlanan yazılımların, AK ve ÖY tanımlarına uygun yazılımlar olmasına önem verildiği belirtilmiştir. Sayfada, son güncelleme tarihi 12 Aralık 2008 itibariyle 247 proje alfabetik sıraya göre listelenmiştir. Yazılımları bir ölçüte göre kategorilere ayırmamıştır.

ÖAK haritacılık projelerini içeren ve bu alandaki haberleri duyuran diğer bir web sitesi de <http://www.maptools.org/> dur. İçerdiği ÖAK projeleri kullanım alanlarına göre Tablo 14’de belirtilen kategoriler doğrultusunda gruplandırmıştır.

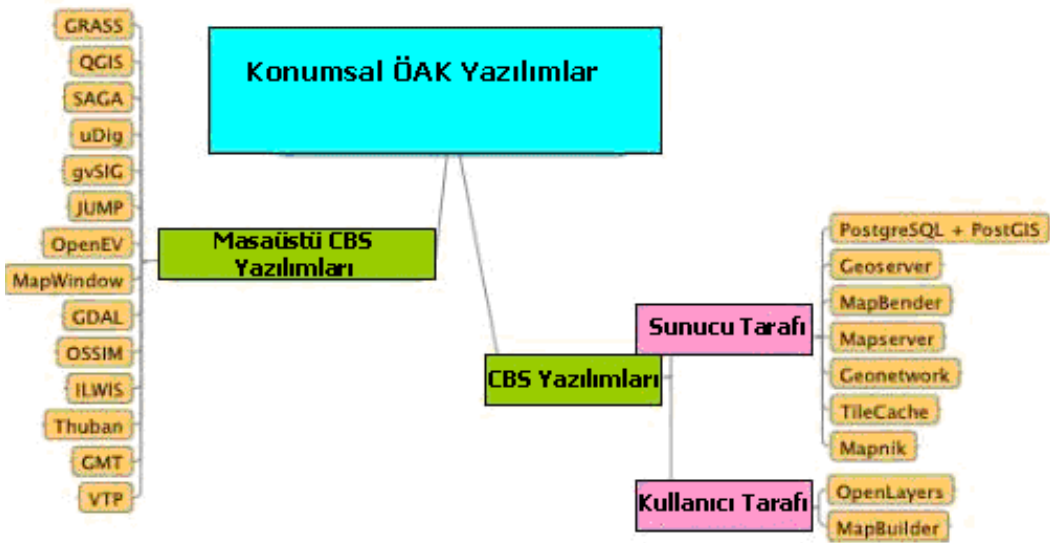
Tablo 14. Maptools.org sitesinin yazılım kategorileri

Kategoriler	Yazılımlar
Araçlar ve Yardımcı Yazılımlar	FGS Linux Installer, FWTools, Mandriva Linux FOSS4G Toolkit, MS4W
Web Araçları	CartoWeb, Chameleon, Fusion, ka-Map, Mapbender, MapFish, MapServer, MapStorer, OpenLayers, OWTChart, PHP MapScript.
Masaüstü Araçlar	GRASS, OpenEV, QuantumGIS, uDig
Yardımcı yazılımlar ve Kütüphaneler	AVCE00, DGNLib, GDAL, GeoTiff, MITAB, OGR, Proj4, Shapelib

Özgür CBS yazılımlarının listelendiği bir diğer bir web sitesi de <http://www.freegis.org/> dur. Sitede haftalık ve aylık olarak son güncellenen projeler belirtilmektedir. Ayrıca, Freegis.org veritabanında, yazılımlar, geo-data (coğrafi veri), dokümanlar ve projeler belli ölçütlere göre gruplandırılmıştır. Bu ölçütler ve gruplar Tablo 15’de gösterilmiştir. Kategorilere göre seçilen yazılımlar ayrıca isim veya önem sırasına göre listelenmektedir.

Tablo 15. Freegis.org sitesindeki yazılım kategorileri

Üst Kategori	Kategoriler
Tür	Yazılım, konumsal –veri, projeler,
Uygulama	Web CBS, Masaüstü CBS, Kütüphane, Veritabanı, Mobil CBS
İşletim Sistemi	GNU/Linux, Windows, MacOS X, diğer işletim sistemleri
Programlama Dili	C, Java, C++, Python, PHP, Tcl/Tk, Perl, Visual Basic, C#, Fortran, Delphy, Ruby, Javascript, SWF, diğer programlama dilleri
Lisans	GNU GPL, GNU LGPL, MIT, Public Domain, BS,Dish, GNU FDL, GPL (v2), Apache Licence, MPL1.1, X 11-Style, GNU GPL v.3, GEOTRANS Lisans, AFL 2.1, Artistic Lisans, Creative Commons, Apache lisans/ non free, AvPython lisans (MIT) , GIS viewer lisansı, GSLIB lisansı, IPW lisansı, kdem lisansı, MapServer lisansı, NCSA-type lisansı, OpenMap Lisansı, partly public domain, QPL, shprans lisansı, SVG toolkit lisansı, vhlmaps lisansı, CeCILL V.2, E00 compr lisansı, zlib lisansı, GNU GPL v.3+ lisansı, GNU GPL v.3+ lisansı , GNU GPLv2+
Özellikler	Veri Dönüşümü, GPS, Geoviewer, koordinat dönüşümü, uzaktan algılama, konumsal analiz, animasyonlar, sayısallaştırma, 3D, yöneltme, Konumsal istatistikler.
Standartlar	OGC WMS client, OGC WFS client, OGC WMS sunucusu, OGC WFS sunucusu, GeoJSON,
Durumu	Aktif olmayan, arşivleme, OsGeo projesi, Debian paketi.



Şekil 17. Örnek konumsal ÖAK yazılımlar (Crăciunescu, vd., 2007)

Şekil 17’de (Crăciunescu, vd., 2007) yaptıkları çalışmada kullanılan ÖAK CBS yazılımları örnek olarak gösterilmiştir.

Araştırma sonuçlarında görüldüğü gibi konumsal ÖAK alanında birçok yazılım bulunmaktadır.

Çoğu zaman aynı yazılımlar farklı kategoriler altında birçok sitede yayınlanmaktadır. Bu durum, “hangi yazılım?” ve “nasıl karar verilebilir?” sorularını gündeme getirmiştir.

2.1.2.2.2. Nihai Seçim

Bu aşamada uygulamada kullanılan yazılım, mevcut yazılımlar arasından belirlenen ölçütlere göre seçilmiştir. Bu ölçütler aşağıda belirtilmiştir.

- İşlevsel kapsam: Bu ölçüt, yazılımın sahip olduğu rutinlerin fazlalığını ve ne kadar “özel” olduğunu belirtmektedir. Buradaki “özel” rutine örnek olarak, uygulama kapsamında belirlenen “parselGeometriGetir” servisinin yaptığı işlem verilebilir. Bir yazılımın işlevsel kapsamı ne kadar fazlaysa aranan rutin bulunma olasılığı da o kadar artmaktadır.
- İşletim Sistemi: Uygulamada kullanılan işletim sistemi Windows XP’ dir. Bu nedenle belirlenecek yazılımın Windows XP’ ye uyumlu çalışan bir yazılım olması tercih edilmiştir.
- Kodun olgunluğu: Kodun olgunluğu kullanılacak yazılımdaki hataların “bug” az olması açısından önemlidir. Yazılım kodu yeterince olgunlaşmış “code maturity” olan yazılımlar tercih edilmiştir.
- Lisans: LGPL lisanslı yazılımlar, ticari yazılımlara da entegre edilebilmeleri yönünden GPL lisansına göre daha esnek olduğu için tercih sebebi olmuştur.
- Programlama Dili: Önceki çalışmalarımızda java kullanıldığı için, programlama dili olarak java tercih edilmiştir.

Çalışma kapsamında incelenen yazılımlara, uygulama konusuna benzer projelerde (Aware, GeOnAs) kullanılan yazılımlar ve yapılan araştırmalar göz önünde bulundurularak karar verilmiştir. Uygulama kapsamında, Aware projesinde de kullanılan Sextante ve GeoTools kütüphaneleri, GeOnAs projesi kapsamında yazılan GRASS tabanlı konumsal web servisleri incelenmiştir.

Bu aşamadaki temel amaç; belirlediğimiz web servislerinin yapacağı işlemler için gereken rutinlerin yazılımlarda olup olmadığını tespit etmektir.

2.1.2.2.2.1. Uygulama Kapsamında İncelenen Yazılımlar

2.1.2.2.2.1.1. Sextante

Sextante konumsal alanda Java tabanlı tek rutin kütüphanesidir. İşlevsel kapsam açısından zengindir. Önceki bölümde de açıklandığı üzere, Sextante kütüphanesinde birçok raster ve vektör veriye yönelik rutin bulunmaktadır. Uygulama vektör verilerle yapılacağı için, Sextante'nin vektör tabanlı rutinleri incelenmiştir. “Clip” (ayırma, kesme) rutininin, web servislerimizden biri olan “binaGeometriGetir” servisinin işlemini yerine getirebileceği düşünülmüştür. “clip”, coğrafi bir bölge içersinden, sadece belli sınırlar çerçevesindeki yeni bir alanın elde edilmesi için kullanılan konumsal bir analiz yöntemidir. Sadece bir parsel sınırlarının içersinde kalan binaların elde edilmesi “clip” analizine örnek olarak verilebilir.

Bu nedenle, “clip” rutinini test etmek üzere Sextante kütüphanesi gvSIG yazılımına entegre edilerek kurulmuştur. Bu işlemde, kesen katman “parsel” ve kesilecek katman ise “bina” katmanıdır.

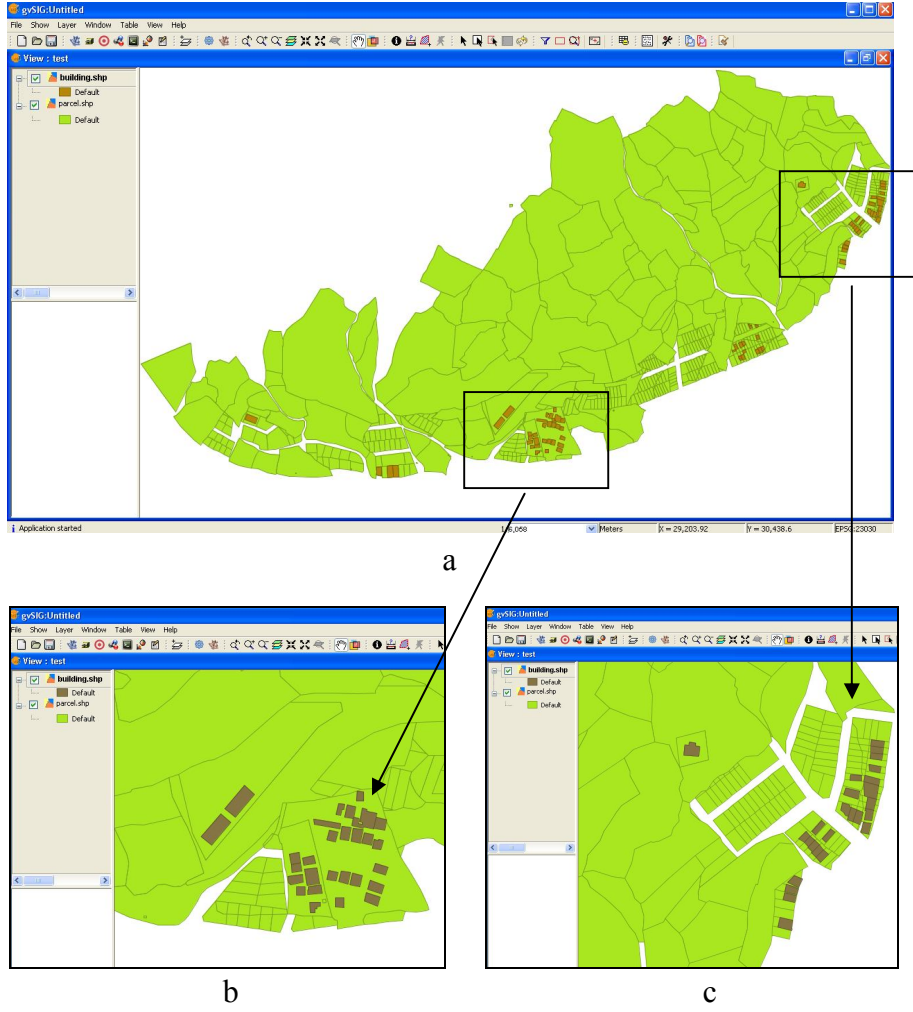
Sextante kütüphanesi ve gvSIG yazılımındaki “clip” rutinlerinin sonuçlarını karşılaştırmak için, her iki yazılımdaki rutinler örnek verilerle denenmiştir.

İşlem sonuçlarının inceleneceği bölgeleri, Şekil 18.b (1. Bölge) ve Şekil18.c (2. Bölge), belirtmek için Şekil 18’ de parsel ve bina katmanları bir arada gösterilmiştir.

Örnek verilerimizle “clip” işlemini gerçekleştirdiğimizde, Sextante “clip” rutininden elde ettiğimiz sonucun doğru olmadığı anlaşılmıştır. Şekil 19.a ve Şekil 19.b’de Sextante “clip” rutininin sonuçları gösterilmiştir.

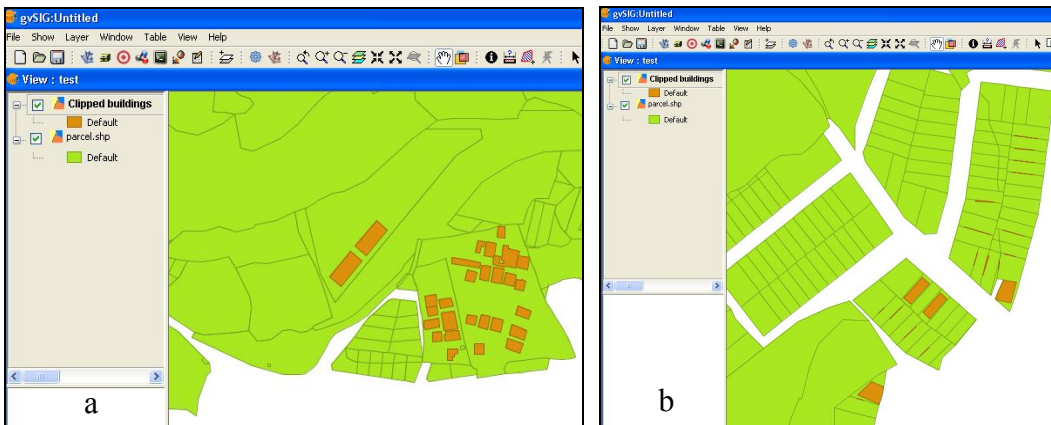
Doğru sonuç, gvSIG yazılımındaki “clip” rutininin sonucundaki gibi olması gerekirken Sextante’deki “clip” rutini doğru bir sonuç vermemiştir. Şekil 20.a ve Şekil 20.b’de gvSIG yazılımının “clip” rutininden elde edilen sonuçlar gösterilmiştir.

Ölçütlerimizden “kodun uygunluğu” ölçütümüze uymadığı için, Sextante yazılımının kullanılmamasına karar verildi.

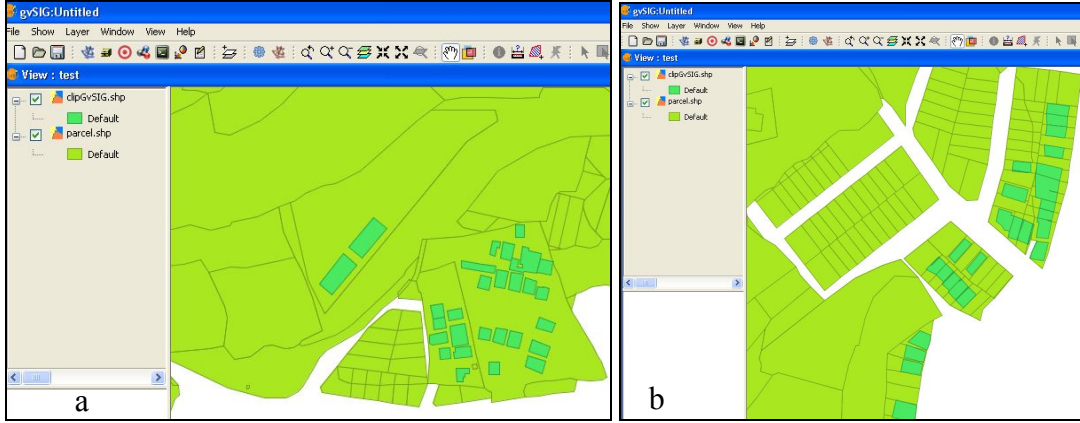


Şekil 18. a) Parsel katmanı ve bina katmanı, b) 1. bölge, c) 2.bölge

Sextante ve gvSIG yazılımlarındaki “Clip” işleminin sonuçlarını rahat karşılaştırmak için sonuçlar iki bölge üzerinde gösterildi.



Şekil 19. a) Sextante “clip” sonucu (1. bölge), b) Sextante “clip” sonucu (2. bölge)



Şekil 20.a) gvSIG “clip” sonucu (1.Bölge), b)gvSIG “clip” sonucu (2.Bölge)

2.1.2.2.1.2. GeOnAs GRASS Tabanlı Web Servisleri

GeOnAs projesi kapsamında geliştirilen konumsal web servislerinin işlevleri ve bu servislerin uygulama kapsamında kullanılıp kullanılmayacağı araştırılmıştır. GRASS ÖAK CBS uygulamalarının içerisinde en eski ve işlevsel kapsamı en fazla olan masaüstü CBS uygulamasıdır. GeOnAs projesinde GRASS yazılımının fonksiyonları kullanılarak konumsal web servislerinin yazıldığından daha önce de bahsedilmiştir. Bu aşamada, GRASS tabanlı konumsal web servislerinin işlevleri araştırılmıştır.

Bu araştırmaya göre, GeOnAs web servislerinden “feature extraction” servisinin belirlediğimiz servislerden, “parselGeometriGetir” veya “binaGeometriGetir” servislerinin işlemlerini yerine getirebileceği görüldü. Fakat bu aşamada da servislerin sadece Linux işletim sisteminde çalışacak şekilde geliştirilmiş olması, işletim sistemi ölçütümüze uymadığı için, bu servislerin de kullanılmamasına karar verildi.

2.1.2.2.1.3. GeoTools

GeoTools kütüphanesini belirlenen ölçütler açısından ele alındığında, işlevsel kapsam açısından Sextante gibi “özel” rutinleri bulunmamaktadır. Fakat JTS ve GeoAPI gibi güçlü kütüphaneler üzerine kurulu olması nedeniyle genel analiz işlemlerini yerine getirebilecek sınıfları bulundurmaktadır. Aktif geliştiricilerinin ve kullanıcılarının geniş bir çerçevede olması, Sextante’ ye göre çok daha uzun süredir geliştiriliyor olması karşılaşılan birçok hatanın daha hızlı bir şekilde çözülmesi anlamına gelmektedir. Bu durum “kodun

olgunluğu” açısından GeoTools kütüphanesinin Sextante’ den daha iyi bir durumda olduğunu göstermektedir. GeoTools kütüphanesi, belirlediğimiz işletim sistemi, programlama dili ve lisans ölçütlerini de sağlamaktadır.

Birçok raster ve vektör veri kaynağıyla işlem yapmasını sağlayan pluginlerinin olması tercih edilmesinde belirlediğimiz ölçütlerin yanı sıra artı değerler olmuştur.

Sonuç olarak, belirlediğimiz ölçütlere uygun olmasından ötürü, uygulama kapsamında GeoTools kütüphanesinin kullanılmasına karar verilmiştir.

2.1.2.3. Servis Kodlarının Geliştirilmesi ve Koşturulması

Öncelikle, uygulama kapsamında belirlenen yazılımı kullanmak ve geliştirilen servisleri koşturmak için gereken programlar belirlenmiştir. Belirlenen servislerden bir kısmı geliştirilmiştir. Geliştirilen web servisleri, uygulama sunucusunda koşturulmuştur.

2.1.2.3.1. Uygulamada Kullanılan Yazılımlar

GeoTools Java tabanlı bir kütüphane olduğu için ön gereksinim olarak Java Development Kit (JDK) kurulmuştur. Bu kapsamda, GeoTools kütüphanesiyle servis kodu geliştirmek için Java uygulama geliştirme ortamı (Java IDE) gereklidir. Uygulamada, GeoTools’ un önerdiği Eclipse ve NetBeans Java IDE’ lerinden, Eclipse tercih edilmiştir. Zorunlu olmamakla birlikte GeoTools Kütüphanesi birçok “.Jar” (Java archive) dosyasından oluştuğu için GeoTools’un önerdiği proje yönetim yazılımı olan Apache Maven kullanılmıştır.

Servislerin koşturulması için, WSO2 WSAS (Web Service Application Server) yazılımı kullanılmıştır.

2.1.2. 3.1.1. J2SDK (Java Software Development Kit)

Java, Sun Microsystems tarafından platformdan bağımsız uygulamalar geliştirmek için, geliştirilen programlama dilidir. Java 1.2 sürümünden sonra, Java2 olarak adlandırılmaya başlanmıştır. Java uygulamaları Java sanal makinesi (JVM) üzerinde çalışır. JVM, üzerinde çalıştığı işletim sistemi ile Java uygulamasının iletişimlerine aracılık eder. JVM için Java Çalışma Ortamı (Java Runtime Environment – JRE) kurulması

gerekmektedir. Java programlarını geliřtirmek için JRE de dahil olmak üzere, gereken temel araçlar Java Uygulama Geliřtirme Takımı (Java 2 Software Development Kit - J2SDK) içersinde bulunmaktadır (Güngören, 2003).

GeoTools Java tabanlı bir kütüphane olduğundan, öncelikle uygulama için GeoTools'un uyumlu çalıştığı Java Jdk 1.6.0 sürümü kurularak gerekli classpath ayarları yapılmıştır.

2.1.2.3.1.2. GeoTools Kütüphanesi

Servis kodunu yazmak için, GeoTools kütüphanesinin 2.5.1 sürümü kullanılmıştır. GeoTools 2.5.1 kütüphanesinin binary "ikili" kodları, <http://sourceforge.net/projects/geotools/files/> adresinden indirilmiştir (Şekil 21). GeoTools sürekli gelişen yapıda olduğu için bir sürümünde olan bir sınıf bir sonraki sürümde iptal edilmiş olabilmektedir.

Örneğin; GeoTools 2.4 sürümünde kendi coğrafi nesne modelini kullanılırken, 2.5 sürümünden sonra coğrafi nesne modeli için GeoAPI kütüphanesinin uyguladığı SFS (simple features) modelini kullanmaya başlamıştır.

2.1.2.3.1.3. Eclipse

Servis kodunu yazmak için uygulama geliştirme ortamı olarak, GeoTools kütüphanesinin de önerdiği açık kaynak kodlu bir yazılım olan, Eclipse Europe Java IDE' si kullanılmıştır.

Eclipse projesi ilk olarak 2001'de IBM'in öncülüğünde geliştirilmeye başlanmıştır. 2004 yılında topluluk (community) projesi olarak ÖAK bir yazılıma uygun olarak geliştirilmeye başlanmıştır. Yalnızca Java IDE'si geliřtirmemektedir. Web uygulama geliştirme araçları, C ve C++ için IDE'ler, mobil java geliřtiriciler için araçlar da yine Eclipse projesi kapsamında geliřtirilmektedir (URL-18, 2009). Şekil 22'de Eclipse Java IDE'si kullanıcı ara yüzü gösterilmiştir.

2.1.2.3.1.4. Yazılım Proje Yönetme Aracı

Apache Maven, Java tabanlı projelerin geliřtirilmesini kolaylařtırmak için geliřtirilen, yazılım projesi yönetme aracıdır (URL-19, 2009).

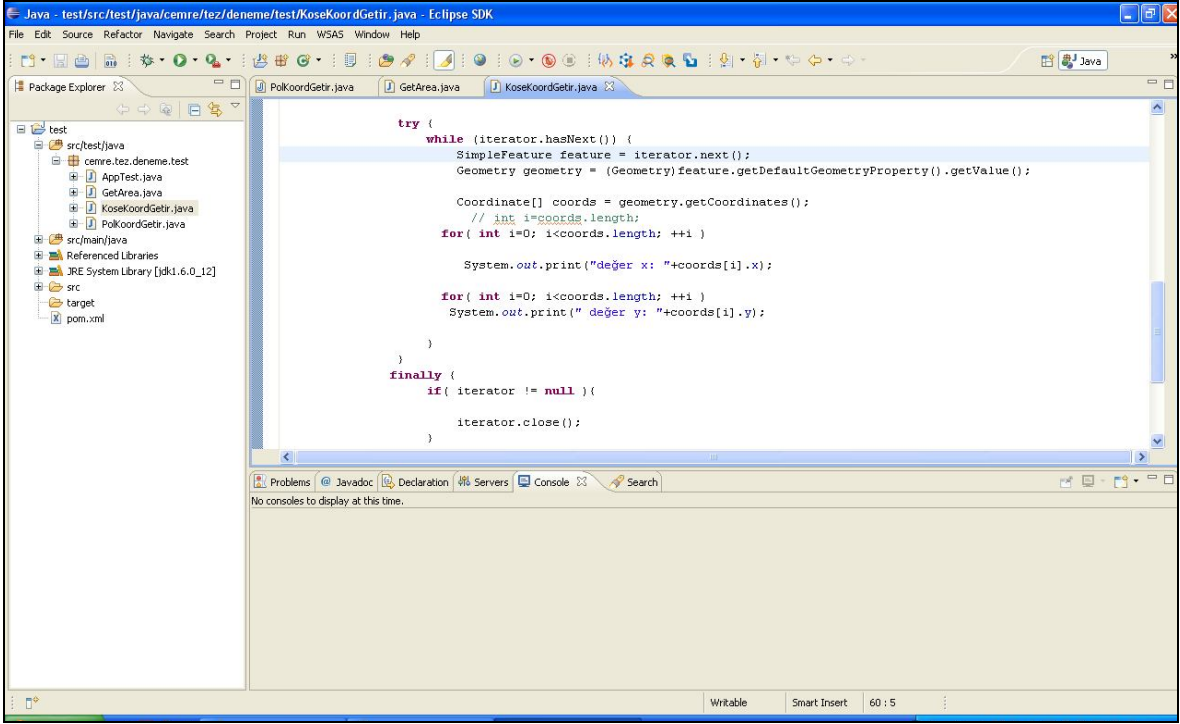
GeoTools kütüphanesi, birçok açık kaynak kodlu kütüphaneden (JTS, GeoAPI) oluştuğu için geniş bir kütüphanedir ve birçok “.jar” (Java Archive) dosyasına sahiptir. Bu “.jar” dosyalarında java sınıfları ve metotları bulunmaktadır. Bir sınıfın çalışması için gerekli diğer sınıflar, farklı “. Jar” dosyalarında olabilmektedir. Bu nedenle, gerekli bütün “.jar” dosyalarının proje “classpath” inde tanımlı olması gerekmektedir. Çok geniş bir kütüphane olan GeoTools, projede sadece gerekli “. Jar” dosyalarıyla çalışılmasını sağlamak için “Apache Maven” – yazılım proje yönetme aracı - kullanılmasını tavsiye etmiştir.

Apache Maven’ in GeoTools ile çalışırken yaptığı en büyük katkı, proje için gerekli “.jar klasörlerini” belirtilen “repository” lerden indirmesidir. Bu sayede kullanıcının yapması gereken işlemler için gerekli temel “.jar” dosyalarını belirtmesi yeterlidir. Buna bağlı olarak yazılımda ihtiyaç duyulan diğer bütün “.jar” dosyaları Maven tarafından indirilmektedir. Uygulamada, Apache maven 2.0.9 sürümü kullanılmıştır.

The screenshot shows the SourceForge project page for GeoTools. The page title is "GeoTools, the java GIS toolkit". Below the title, there is a description: "GeoTools is an open source java GIS toolkit. Used for OGC based projects via GeoAPI interfaces. Includes two great SLD based renderers, raster access and reprojection. Plugins fo Shapefile, ArcGrid, ArcSDE, Postgis, OracleSpatial, MySQL and many more." Below the description, there is a table titled "Browse Files for GeoTools, the java GIS toolkit". The table has the following columns: File/Folder Name, Platform, Size, Date, Downloads, and Notes/Subscribe. The table lists several files, with the newest file being "geotools-2.6-M2-project.zip" (46.1 MiB, Thu Sep 03 2009 06:48, 894 Downloads). Below the table, there are sections for "All Files" and "GeoTools 2.2 Releases", "GeoTools 2.3 Releases", "GeoTools 2.4 Releases", "GeoTools 2.5 Releases", and "GeoTools 2.6 Preview".

File/Folder Name	Platform	Size	Date	Downloads	Notes/Subscribe
Newest Files					
geotools-2.6-M2-project.zip		46.1 MiB	Thu Sep 03 2009 06:48	894	
All Files					
GeoTools 2.2 Releases					
▶ 2.2.0					
▶ 2.2.1					
▶ 2.2.2					
GeoTools 2.3 Releases					
GeoTools 2.4 Releases					
GeoTools 2.5 Releases					
GeoTools 2.6 Preview					

Şekil 21. <http://sourceforge.net/projects/geotools/files/> ara yüzünün ekran görüntüsü



Şekil 22. Eclipse Java IDE kullanıcı ara yüzü

2.1.2.3.1.5. Web Servisleri İçin Uygulama Sunucusu

WSO2 WSAS (Web Services Application Server) Axis2 tabanlı web servisleri için açık kaynak kodlu, bir uygulama sunucusudur. WSO2, servis yönelimli mimariye yönelik web servisleri araçları geliştirmektedir (URL–20, 2009).

Uygulama kapsamında geliştirilen web servislerinin koşturulması için WSO2 WSAS 2.3 sürümü kullanılmıştır.

2.1.2.3.2. Uygulamada Geliştirilen Konumsal Web Servisleri

Uygulama kapsamında belirlenen konumsal web servislerinden, ParselAlanHesapla servisi, KöşeKoordinatGetir servisi, PoligonKoordinatGetir servisi, ParselCepheHesapla servisi ve BinaKöşeKoordinatGetir servisleri yazılmıştır. Uygulama kapsamında, “shapefile” formatında veriler kullanılmıştır. Bu veriler; “parsel.shp” , “poligon.shp”, “bina.shp” dir.

Şekil 23’de web servis kodlarının Eclipse uygulama ara yüzündeki görüntüsü gösterilmiştir.

Örnek olarak “ParselAlanHesapla” kodunun sonuç ekran görüntüsü Şekil 24’de gösterilmiştir.

Geliştirilen konumsal web servislerine ait kodlar tezin “Ekler” bölümünde belirtilmiştir.

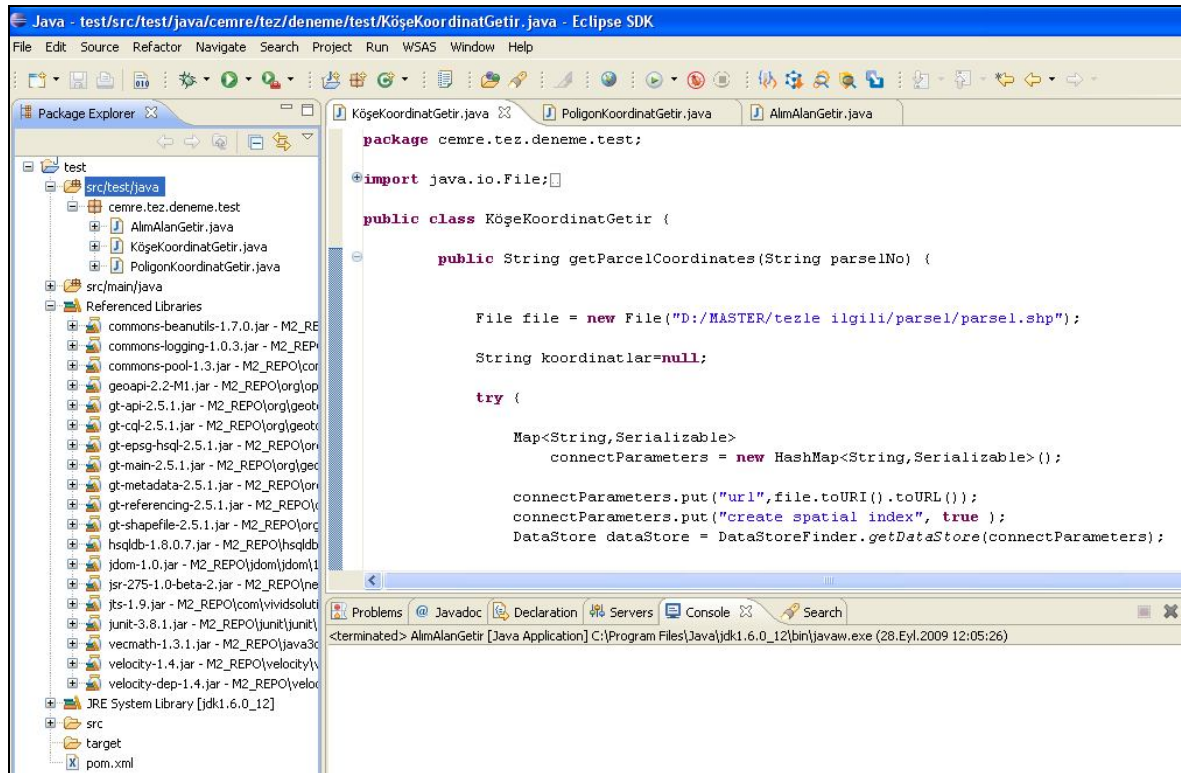
ParselAlanHesapla Servisi: Girdi parametresi olarak, “parsel id” girilir. Çıktı olarak, “parsel id” si girilen parselin alanını döndürür.

KöşeKoordinatGetir Servisi: Girdi parametresi olarak, “parsel id” girilir. Çıktı olarak “parsel id” si girilen parselin köşe koordinatlarını döndürür.

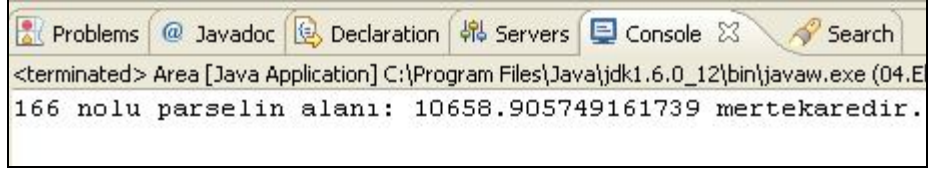
PoligonKoordinatGetir Servisi: Girdi parametresi olarak, “parsel id” girilir. Çıktı olarak parsele yirmi metre mesafede bir buffer bölgesi oluşturularak bu buffer bölgesi içerisindeki poligon noktalarının koordinatları ve poligon numaralarını döndürür.

ParselCepheHesapla Servisi: Girdi parametresi olarak, “parsel id” girilir. Çıktı olarak parselin cephe uzunluklarını döndürür.

BinaKöşe KoordinatGetir: Girdi parametresi olarak “parsel id” girilir. Çıktı olarak parselin üzerinde bina bulunup bulunmadığını kontrol eder ve eğer bina varsa, bu binanın köşe koordinatlarını döndürür.



Şekil 23. Eclipse uygulama geliştirme ara yüzü, geliştirilen web servis kodları



```
<terminated> Area [Java Application] C:\Program Files\Java\jdk1.6.0_12\bin\javaw.exe (04.E
166 nolu parselin alanı: 10658.905749161739 mertekaredir.
```

Şekil 24. ParselAlanHesapla kodunun sonuç ekran görüntüsü

3. BULGULAR VE İRDELEME

Bu tez çalışmasındaki temel amaç, Konumsal Veri Altyapılarına (KVA) yönelik web servislerinin konumsal Özgür ve Açık Kaynak kodlu (ÖAK) yazılımlardaki uygulanmış rutinlerle geliştirilmesidir.

Çalışmada konumsal ÖAK yazılımlar araştırılmıştır. Konumsal ÖAK yazılımlar web sitelerinde duyurulmaktadır. Her web sitesi farklı ölçütlere göre yazılımları sınıflandırmakta ve duyurdukları yazılım sayısı bakımından birbirleri arasında ciddi farklılıklar bulunmaktadır. Bazı yazılımlar birden çok sitede farklı kategoriler altında duyurulmaktadır.

Bu kadar çeşitli yazılımların arasından “hangi yazılımın” “nasıl” seçileceği sorusuna yanıt aranmıştır. Bu nedenle uygulamada kullanılacak yazılım veya yazılımları belirlemek için bazı “ölçütler” belirlenmiştir.

Belirlenen ölçütler kapsamında, uygulama sırasında edinilen deneyim ve yapılan araştırmaların sonucunda, konumsal ÖAK yazılımların bazı eksiklikleri, ticari yazılımlara göre dezavantajları ve avantajları bulunduğu belirlenmiştir. Esasında, avantajlar ve dezavantajlar konusunda bütün konumsal ÖAK yazılımları içine alacak şekilde bir genelleme yapmak çok doğru bir yaklaşım değildir. Bu avantajlar yazılımdan yazılıma değişiklik göstermektedir. Örneğin, İyi finansal destek gören, geliştiriciler ve kullanıcılar açısından zengin, “kod olgunluğu” iyi durumda olan bir yazılımın, karşılaşılan sorunların çözülmesi konusunda kullanıcılarına sağladığı hizmet, bir ticari yazılıma göre çok daha iyi seviyede olmaktadır. Fakat bunun tam tersi bir durum söz konusu olduğunda ticari yazılımlar bu açıdan daha avantajlı olmaktadır.

Uygulama kapsamında konumsal ÖAK yazılımların ticari yazılımlara göre belirlenen avantajları aşağıda anlatılmıştır.

- Çoğunlukla ücretsiz ya da nadiren çok az bir ücret karşılığında dağıtılmaları; Ticari yazılımlarda ücretler lisanslara ödenmektedir. Ticari firmalar yazılımların lisanslarına belli bir süre için kullanma izni vermektedirler. Bu uygulama, ticari yazılımların ekonomik açıdan külfetli olmasına sebep olmaktadır. ÖAK yazılımlar kullanıcıların “yazılım kullanma özgürlüğü” ilkesiyle geliştirildikleri için, genellikle ücretsiz olarak dağıtılmaktadır.

Uygulamada kullanılan GeoTools yazılımı ücretsiz dağıtılan bir ÖAK yazılımdır. Bu sayede uygulamada yazılım edinilirken hiçbir ücret sarf edilmemiştir.

- Kullanıcılar ve geliştiricileri ile aktif olarak iletişim kurabilecekleri ortamların bulunması; Eğer kullanılan ÖAK yazılım belirli bir kullanıcı ve geliştirici topluluğu oluşturabilmişse karşılaşılan sorunlara, yazılımın e-posta listeleri, IRC görüşme ortamları gibi platformlarda çok kısa bir süre içerisinde yanıt bulunmaktadır. Fakat aktif bir ortam oluşturamayan yazılım topluluklarında bu hizmetin çok fazla faydası olmamaktadır.

Çalışma sırasında Sextante yazılımının geliştiricileriyle karşılaşılan sorunlar ile ilgili yazışmalar yapılarak sorun bildirilmiştir. Geliştiriciler bir kaç hafta içinde cevap verileceğini belirtmişlerdir. Ancak aradan üç ay geçmesine rağmen hala bir cevap alınamamıştır.

GeoTools kütüphanesiyle web servislerini geliştirirken karşılaşılan sorunlara cevap bulmak için GeoTools e-posta listelerinden aktif bir şekilde faydalanılmıştır. Her soruya karşılık, bazen iki saat gibi çok kısa bir zaman içerisinde cevap alınmıştır.

- Kaynak kodun açık olması; ÖAK yazılımların ticari yazılımlara göre sağladığı en büyük avantajdır. ÖAK yazılımlarının kaynak kodlarının okunabilmesi, güvenlik açısından büyük önem taşımaktadır. Bunun yanı sıra o yazılımla yapılan işlemlerin arka planlarının anlaşılması, yeni fikirler geliştirilmesi açısından da büyük fayda sağlamaktadır.

Uygulama kapsamında konumsal ÖAK yazılımların ticari yazılımlara göre belirlenen dezavantajları aşağıda anlatılmıştır.

- Kod için çalışma garantisinin verilmemesi; ÖAK yazılımlar GPL lisansının şartları gereğince, yazılımın kalitesi veya performansı ile ilgili sorumluluk kabul etmemektedir. Yazılımda herhangi bir bozukluktan dolayı doğabilecek olan bütün servis, tamir veya düzeltme masraflarını kullanıcıya bırakmaktadır. Bu durumda kullanıcıların yazılımı kullanırken karşılaştıkları sorunlar ve bunun sonucunda doğacak zararlar için hiçbir kişi veya kurumu sorumlu tutamamaktadır.
- Yeterince kontrol edilmeden yayınlanması; Her ne kadar ÖAK yazılımlar lisanslarında yazılımın doğru çalışması için bir garanti vermiyor olsalar da, ÖAK yazılımlarının kullanımını teşvik etmek ve güven kazandırmak açısından,

yazılımların yayınlanmadan önce daha çok kontrol edilmesi gerektiği belirlenmiştir.

Uygulama sırasında Sextante rutin kütüphanesindeki “clip” rutininin yanlış çalıştığı halde yayınlanmış olması bu eksikliğe örnek verilebilir.

- Dokümantasyonlarının yeterli, güncel ve anlaşılır olmaması; Yazılımların anlaşılması ve uygulamada zorluk yaşanmaması açısından dokümantasyon büyük önem taşımaktadır. ÖAK yazılımları büyük topluluklar “community” ler tarafından sürekli gelişim halinde olduğu için, çok sık güncellenmektedir. Bu durumda aktif kullanımını her düzeyden kullanıcıya yaymak için dokümantasyon işleminin güncel olması ÖAK yazılımlar için önem kazanmaktadır.

Uygulamada kullanılan GeoTools Kütüphanesi birçok farklı kütüphaneden oluşmaktadır. Buna bağlı olarak karmaşık ve sürekli güncellenen bir yazılım kütüphanesidir. Fakat dokümantasyonunun yeterli ve güncel olmaması yazılımın çalışma prensibinin çok iyi anlaşılmasına ve kullanımında zorluk yaşanmasına sebep olmuştur.

Çalışma sırasında en çok zorluk yaşanan süreç, Web servislerinin yapacağı işlemi gerçekleştiren bir rutin, mevcut yazılımlar kapsamında uygulanıp uygulanmadığını anlama süreci olmuştur.

ÖAK yazılımlarının kaynak kodlarının açık olması nedeniyle bu kodlar kullanıcı ihtiyaçlarına göre değiştirilebilir veya aynı işi yapacak bir rutin uygulanmışsa yeniden yazılmasına gerek kalmadan kullanılabilir. Bu özellik, ÖAK yazılımlar için iyi bir avantaj olarak gözükse de uygulama aşamasında bazı eksikliklerin olduğu belirlenmiştir. Bu aşamada aranılan rutin yazılımda olup olmadığının anlaşılması için “arama mekanizmaları” veya benzeri pratik bir yöntem bulunmamaktadır.

Şu anki sistemde yazılımların işlevsel kapsamı ancak yazılımı anlatan dokümanlar veya yazılımın kaynak kodu incelenerek öğrenilmektedir. Bu yöntemin, mevcut yazılımların sayısı göz önünde bulundurulduğunda çok fazla vakit alan, araştırma sürecini zorlaştıran ve pratik olmayan bir yöntem olduğu görülmüştür.

4. SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında, konumsal alandaki Özgür ve Açık Kaynak kodlu (ÖAK) yazılımların Konumsal Veri Altyapıları (KVA) için Web servislerinin gerçekleştirilmesinde kullanılıp kullanılmayacağı araştırılmıştır.

Çalışmanın ilk aşamasında, konumsal ÖAK yazılımlar araştırılmış, birçok yazılım arasından uygulamada kullanılacak yazılımı belirlemek için bazı ölçütlere ihtiyaç duyulmuştur. Bu ölçütler doğrultusunda uygulamada kullanılan yazılım, GeoTools kütüphanesi, belirlenmiştir.

Çalışmanın ikinci aşamasında GeoTools yazılımı kullanılarak örnek uygulama için belirlenen web servislerinden bir kısmı yazılmıştır.

Uygulama sürecinde ÖAK yazılımlarının kullanılması sonucunda, konumsal ÖAK yazılımlarının ticari yazılımlara göre avantajları ve dezavantajları belirlenmiştir.

Belirlenen avantajlar: Kaynak kodunun açık olması, çoğunlukla ücretsiz veya nadiren çok az bir ücret karşılığında dağıtılmaları, kullanıcılar ve geliştiricileri ile aktif olarak iletişim kurabilecekleri ortamların bulunmasıdır.

Belirlenen dezavantajlar: Kod için çalışma garantisinin verilmemesi, ÖAK yazılımlarının yeterince kontrol edilmeden yayınlanması, dokümantasyonlarının yeterli, güncel ve anlaşılır olmamasıdır.

Bunların yanı sıra ÖAK yazılımların kaynak kodunun açık olmasından faydalanmak, uygulama kapsamında bahsedecek olursak, uygulanan rutini yeniden kullanmak için, bu rutinlerin kaynak kodda bulunup bulunmadığının araştırılması gerekmektedir. Fakat yazılımın içeriğini araştırmak için herhangi bir “arama mekanizması” bulunmamaktadır. Bu kapsamda, ÖAK yazılımlarının böyle bir mekanizma içermesi, konumsal alandaki uygulama geliştiricilere büyük bir kolaylık sağlayacaktır.

Ülkemizde konumsal yazılımlar açısından büyük oranda dışa bağımlılık söz konusudur. Bu durum ülkemize dışa bağımlılığın yanı sıra maddi yük de getirmektedir. Bu durumun ortadan kaldırılmasında en büyük sorumluluk eğitim kurumlarına düşmektedir. Üniversiteler, yalnızca kullanıcı bireyler değil üretici bireyler de yetiştirebilmek için, konumsal ÖAK yazılımlarını eğitim sistemlerine adapte etmelidir.

5. KAYNAKLAR

- Akıncı, H., 2006. Konumsal Veri Altyapılarının Web Servisleri ile Gerçekleştirilmesi: İçin Mevcut Durum Analizi ve Gelecek Yönelimlerinin Belirlenmesi , Doktora Tezi, KTÜ, Fen Bilimleri Enstitüsü, Trabzon.
- Crăciunescu, V., Constantinescu, Şt. ve Ovejanu, I., 2008. Developing an Open Romanian Geoportal using free open source software, *Gegraphia Technica*, No. 1, Cluj University Press, 15–20.
- Di, L., GeoBrain-A Web Services based Geospatial Knowledge Building System, Earth Science Technology Conference, Haziran, Palo Alto (CA,USA), http://geobrain.laits.gmu.edu/papers/Geobrain_Webservice.pdf, 18 Eylül 2009.
- Di, L., Zhao, P., Han, W., Wei, Y. ve Li, X., Web Service-based GeoBrain Online Analysis System (GeOnAS), Proceedings of NASA Earth Science Technology Conference, Haziran, Collage Park (MD, USA), <http://geobrain.laits.gmu.edu/papers/nstc2007.pdf>, 15 Eylül 2009.
- Díaz, L., Costa, S., Granell, C. ve Gould, M., Migrating geoprocessing routines to web services for water resource management applications, 10th AGILE International Conference on Geographic Information Science, Mayıs, Aalborg (Denmark), http://people.plan.aau.dk/~enc/AGILE2007/PDF/151_PDF.pdf, 12 Eylül 2009.
- Díaz, L., Granell, C., Gould, M. ve Olaya, V., An Open Service Network for Geospatial Data Processing, 2008 Free and Open Source Software for Geospatial Conference, Cape Town, South Africa, <http://conference.osgeo.org/index.php/foss4g/2008/paper/viewPDFInterstitial/301/98>, 16 Eylül 2009.
- Díaz, L., Granell, C. ve Gould, M., 2008. Case Study: Geospatial Processing Services for Web-based Hydrological Applications, *Geospatial Services and Applications for the Internet*, Springer Science+Business Media, ISBN 978-0-387-74673-9, 31-47.
- Garnett J. ve Bedward M.,
Render, <http://docs.codehaus.org/display/GEOTDOC/12+Render>, 10 Eylül 2009.
- Garnett, J., Meet the GeoTools Library,
<http://docs.codehaus.org/display/GEOTDOC/02+Meet+the+GeoTools+Library>, 20 Ağustos 2009.
- Garnett, J., GeoAPI, <http://docs.codehaus.org/display/GEOTDOC/02+GeoAPI>, 20 Ağustos 2009.
- Garnett, J., Java Topology Suite, <http://docs.codehaus.org/display/GEOTDOC/03+JTS+Topology+Suite>, 20 Ağustos 2009.

- Gözükeleş, İ., İ., 2004. Özgür/Açık Kaynak Kodlu Yazılım ve Ulusal Yazılım Politikaları.BM-Dergisi, 34-42.
- Güngören, B., 2003. Java ile Temel Programlama, Seçkin Yayıncılık, Ankara, ISBN: 975 347 616 7, 437 s.
- Hall, G.,B. ve Leahy, M.,G., 2008 Open Source Approaches in Spatial Data Handling, Springer, 283 s.
- Han, W., Di, L., Zhao, P., Wei, Y. ve Li, X., Design and Implementation of GeoBrain Online Analysis System, W2GIS 2008 Symposium, Aralık, Shanghai (China), <http://www.springerlink.com/index/c303j50701742358.pdf>, 3 Ekim 2009.
- Kepoğlu, V.,O. ve Düzgün, Ş., Özgür CBS Yazılımı Geliştirme: Quantum GIS Örneği, 2. Uzaktan Algılama ve Coğrafi Bilgi Sistemleri Sempozyumu, Ekim, Kayseri, <http://ggit.metu.edu.tr/~volkan/OZGUR%20CBS%20YAZILIMI%20GELISTIRME%20QUANTUM%20GIS%20ORNEGI.pdf>, 10 Eylül 2009.
- Lemmens, R., Theodor, F. ve Barend K., 2007. Service Chaining & Interoperability of WMS, WFS and WPS in 52°North Initiative, AGILE INTEROP Workshop, Mayıs, Aalborg (Denmark).
- OGC, 2005 Filter Encoding Implementation Specification, OGC 04-095, OpenGIS Consortium.
- OGC, 2005 Web Processing Service, version 1.0.0, OGC-05007r7, OpenGIS Consortium.
- OGC, 2006 Simple Features for SQL, version 1.2.0, OGC- 06104r3, OpenGIS Consortium.
- Olaya, V., SEXTANTE, A Free Platform for Geospatial Analysis, <https://forge.osor.eu/docman/view.php/13/74/ArticleForOsGEOJournal.pdf> , 17 Eylül 2009.
- Olaya, V., http://forge.osor.eu/plugins/scmsvn/viewcvs.php/*checkout*/docs/LaTeX/en/ProgrammingSextante/ProgrammingGuide.pdf?root=sextante, 17 Eylül 2009.
- Ramsey, P., 2007. The State of Open Source GIS, FOSS4G Conference 2007, Vancouver, http://2007.foss4g.org/presentations/viewattachment.php?attachment_id=8, 15 Eylül, 2009.
- Roldan, G. ve Garnett, J., Source Licence, <http://docs.codehaus.org/display/GEOTDOC/00+Source+License>, 1 Ekim 2009.
- Steiniger, S. ve Bocher, E., 2008. An Overview on Current Free and Open Source Desktop GIS Developments, Int. J. of Geographical Information Science, http://www.geo.unizh.ch/publications/degen/sstein_foss_desktop_gis_overview.pdf, 12 Eylül 2009.
- Ulutaş, D., 2008. Semantik Web Servisleri ve Konumsal Veri Altyapıları , Yüksek Lisans Tezi, KTÜ, Fen Bilimleri Enstitüsü, Trabzon.

Westervelt, J., 2004. GRASS Roots, Free/Libre and Open Source Software for Geoinformatics: GIS-GRASS Users Conference, Bangkok (Thailand),

URL-1, <http://www.gnu.org/philosophy/free-sw.html>, 18 Eylül 2009.

URL-2, <http://www.fsf.org/about/what-is-free-software>, 19 Eylül 2009.

URL- 3, <http://www.gnu.org/philosophy/open-source-misses-the-point.html>, 19 Eylül 2009

URL-4, http://www.fsf.org/licensing/licenses/index_html#GNUGPL, 19 Eylül 2009.

URL-5, <http://52north.org/index.php>, 25 Eylül 2009.

URL-6, http://52north.org/index.php?option=com_content&view=article&id=14&Itemid=16, 15 Eylül 2009.

URL - 7, <http://www.fsf.org/licensing/licenses/lgpl.html>, 20 Eylül 2009.

URL - 8, <http://52north.org/>, 20 Eylül 2009.

URL-9, <http://52north.org/maven/project-sites/wps/52n-wps-webapp/>, 16 Eylül 2009.

URL-10 <http://www.vividsolutions.com/jts/jtshome.htm>, 12 Eylül 2009.

URL-11 <http://docs.codehaus.org/display/GEOTDOC/02+GeoAPI>, 27 Ağustos 2009.

URL-12 <http://geoapi.sourceforge.net/>, 27 Ağustos 2009.

URL-13, <http://docs.codehaus.org/display/GEOTDOC/02+Meet+the+GeoTools+Library>, 19 Ağustos 2009.

URL-14, <http://forge.osor.eu/plugins/wiki/index.php?The%20algorithms&id=13&type=g>, 15 Eylül 2009.

URL-15, <http://forge.osor.eu/plugins/wiki/index.php?Who%20is%20using%20SEXTANTE&id=13&type=g>, 15 Eylül 2009.

URL - 16, <http://grass.itc.it/intro/general.php>, 18 Eylül 2009.

URL- 17, <http://geobrain.laits.gmu.edu:81/grassweb/axis2/> , 4 Ekim 2009.

URL - 18, <http://www.eclipse.org/org/> , 3 Ekim 2009.

URL-19, <http://maven.apache.org/>, 3 Ekim 2009.

URL-20, <http://wso2.org/projects/wsas/java>, 4 Ekim 2009.

6. EKLER

Ek.1 ParselAlanHesapla Servis Kodu

```
package kadastro.servisler.tez.test;

import java.io.File;
import java.io.Serializable;
import java.util.HashMap;
import java.util.Map;

import org.geotools.filter.text.cql2.*;
import org.geotools.data.DataStore;
import org.geotools.data.DataStoreFinder;
import org.geotools.data.FeatureSource;
import org.geotools.feature.FeatureCollection;
import org.geotools.feature.FeatureIterator;

import org.opengis.filter.*;
import org.opengis.feature.simple.SimpleFeature;
import org.opengis.feature.simple.SimpleFeatureType;

import com.vividsolutions.jts.geom.Geometry;

public class ParselAlanHesapla {

    public double getParselAlan(String parcelNo) {

        File file = new File ("D:/MASTER/tezle ilgili/parsel/parsel.shp" );

        double area=0.0;

        try {
            Map<String,Serializable>
            connectParameters = new HashMap<String,Serializable>();

            connectParameters.put("url", file.toURI().toURL());
            connectParameters.put("create spatial index", true );
            DataStore dataStore = DataStoreFinder.getDataStore(connectParameters);

            String[] typeNames = dataStore.getTypeNames();
            String typeName = typeNames[0];

            FeatureSource<SimpleFeatureType, SimpleFeature> featureSource;
            FeatureCollection<SimpleFeatureType, SimpleFeature> collection;
            FeatureIterator<SimpleFeature> iterator;
```

Ek 1'in devamı

```
featureSource = datastore.getFeatureSource(typeName);

Filter filter = CQL.toFilter(("K1_ID="+parcelNo));
collection = featureSource.getFeatures(filter);
iterator = collection.features();

try {

    while (iterator.hasNext()) {
        SimpleFeature feature = iterator.next();

        Geometry geometry = (Geometry) feature.getDefaultGeometry();
        area=geometry.getArea();

        //System.out.print(attribute+ " nolu parselin alanı: "+area);
    }
}
finally {
    if( iterator != null ){

        iterator.close();
    }

}

} catch (Exception ex) {
    ex.printStackTrace();
    System.exit(1);
}
return area;

}
}
```

Ek 2. KöşeKoordinatGetir Servis Kodu

```

package kadastro.servisler.tez.test;

import java.io.File;
import java.io.Serializable;
import java.util.HashMap;
import java.util.Map;

import org.geotools.data.DataStore;
import org.geotools.data.DataStoreFinder;
import org.geotools.data.FeatureSource;

import org.geotools.feature.FeatureCollection;
import org.geotools.feature.FeatureIterator;
import org.geotools.filter.text.cql2.CQL;
import org.opengis.feature.simple.SimpleFeature;
import org.opengis.feature.simple.SimpleFeatureType;
import org.opengis.filter.Filter;

import com.vividsolutions.jts.geom.Coordinate;
import com.vividsolutions.jts.geom.Geometry;

public class KöşeKoordinatGetir {

    public String getParcelCoordinates(String parselNo) {

        File file = new File("D:/MASTER/tezle ilgili/parsel/parsel.shp");

        String koordinatlar=null;

        try {

            Map<String,Serializable>
                connectParameters = new HashMap<String,Serializable>();

            connectParameters.put("url",file.toURI().toURL());
            connectParameters.put("create spatial index", true );
            DataStore dataStore =
DataStoreFinder.getDataStore(connectParameters);

            String[] typeNames = dataStore.getTypeNames();
            String typeName = typeNames[0];

            System.out.println("Reading content " + typeName);

            FeatureSource<SimpleFeatureType, SimpleFeature> featureSource;
            FeatureCollection<SimpleFeatureType, SimpleFeature> collection;

```

Ek 2'nin devamı

```

FeatureIterator<SimpleFeature> iterator;

featureSource = dataStore.getFeatureSource(typeName);

Filter filter = CQL.toFilter(("K1_ID="+parselNo));
collection = featureSource.getFeatures(filter);

iterator = collection.features();

try {
    while (iterator.hasNext()) {
        SimpleFeature feature = iterator.next();
        Geometry geometry =
(Geometry)feature.getDefaultGeometryProperty().getValue();

        Coordinate[] coords = geometry.getCoordinates();

        for(int i=0; i<coords.length; i++){
            koordinatlar=koordinatlar+coords[i].y+" "+coords[i].x+" ";
        }

    }
} finally {
    if( iterator != null ){

        iterator.close();
    }

}

} catch (Exception ex) {
    ex.printStackTrace();
    System.exit(1);
}
return koordinatlar;
}
}

```

Ek.3 PoligonKoordinatGetir Servis Kodu

```

package kadastro.servisler.tez.test;

import java.io.File;
import java.util.HashMap;
import java.util.Map;

import org.geotools.factory.*;
import org.opengis.filter.*;
import org.geotools.filter.text.cql2.*;
import org.geotools.data.DataStore;
import org.geotools.data.DataStoreFinder;
import org.geotools.data.FeatureSource;
import org.geotools.feature.FeatureCollection;
import org.geotools.feature.FeatureIterator;

import org.opengis.feature.simple.SimpleFeature;
import org.opengis.feature.simple.SimpleFeatureType;
import org.opengis.geometry.*;

import com.vividsolutions.jts.geom.Coordinate;
import com.vividsolutions.jts.geom.Geometry;

public class PoligonKoordinatGetir {

    public String getPolCoordinates(String parselNo) {

        File file = new File ("D:/MASTER/tezle ilgili/parsel/parsel.shp" );
        File file2 = new File ("D:/MASTER/tezle ilgili/polygon/poligon.shp" );

        String koordinatlar=null;

        try{

            Map<String,Object> connect = new HashMap<String,Object>();
            connect.put("url", file.toURI().toURL());
            DataStore shapefile = DataStoreFinder.getDataStore(connect);
            String[] typeNames = shapefile.getTypeNames();
            String typeName = typeNames[0];

            FeatureSource<SimpleFeatureType, SimpleFeature> featureSource;
            FeatureCollection<SimpleFeatureType, SimpleFeature> collection;

            featureSource = shapefile.getFeatureSource(typeName);

            Filter filter = CQL.toFilter(("K1_ID="+parselNo));
            collection = featureSource.getFeatures(filter);

```

Ek 3'ün devamı

```

BoundingBox ft=collection.getBounds();
Double j,k,l,m;
j=ft.getMaxX()+20;
k=ft.getMaxY()+20;
l=ft.getMinX()-20;
m=ft.getMinY()-20;

FilterFactory2 ff = CommonFactoryFinder.getFilterFactory2(null);

Filter filter2 = ff.bbox("the_geom", l, m, j, k,null );

Map<String,Object> connect2 = new HashMap<String,Object>();
connect2.put("url", file2.toURI().toURL());
DataStore shapefile2 = DataStoreFinder.getDataStore(connect2);
String typeName2 = shapefile2.getTypeNames()[0];

FeatureSource<SimpleFeatureType, SimpleFeature> fs;
FeatureCollection<SimpleFeatureType, SimpleFeature> cl;
FeatureIterator<SimpleFeature> iterator;
fs=shapefile2.getFeatureSource(typeName2);

cl=fs.getFeatures(filter2);
iterator = cl.features();

try {
    while (iterator.hasNext()) {
        SimpleFeature feature = iterator.next();
        Geometry geometry= (Geometry)feature.getDefaultGeometry();
        Coordinate[] coords = geometry.getCoordinates();
        Object attribute = feature.getAttribute("POL_ID");

for( int i=0; i<coords.length; ++i )
        { coordinatlar=coordinatlar+attribute+" "+coords[i].y+" "+coords[i].x+" ";
          }
        }
    }
    finally {
if( iterator != null ){
        }
    }
    } catch (Exception ex) {
    ex.printStackTrace();
    System.exit(1);
    }
return coordinatlar;
    }
}

```

Ek 4. ParselCepheHesapla Servis Kodu

```

package kadastro.servisler.tez.test;

import java.io.File;
import java.io.Serializable;
import java.util.HashMap;
import java.util.Map;
import java.awt.geom.Point2D;

import org.geotools.data.DataStore;
import org.geotools.data.DataStoreFinder;
import org.geotools.data.FeatureSource;

import org.geotools.feature.FeatureCollection;
import org.geotools.feature.FeatureIterator;
import org.geotools.filter.text.cql2.CQL;
import org.opengis.feature.simple.SimpleFeature;
import org.opengis.feature.simple.SimpleFeatureType;
import org.opengis.filter.Filter;

import com.vividsolutions.jts.geom.*;

public class ParselCepheHesaplaServisi {

    public double [] parselCepheGetir(String parselNo){

        File file = new File("D:/MASTER/tezle ilgili/parsel/parsel.shp");

        double[] sideLength = new double[1];

        try {

            Map<String,Serializable>
            connectParameters = new HashMap<String,Serializable>();

            connectParameters.put("url",file.toURI().toURL());
            connectParameters.put("create spatial index", true );
            DataStore dataStore =
DataStoreFinder.getDataStore(connectParameters);

            String[] typeNameNames = dataStore.getTypeNames();
            String typeName = typeNameNames[0];

            FeatureSource<SimpleFeatureType, SimpleFeature> featureSource;
            FeatureCollection<SimpleFeatureType, SimpleFeature> collection;

```

Ek 4'ün devamı

```

FeatureIterator<SimpleFeature> iterator;

featureSource = datastore.getFeatureSource(typeName);

Filter filter = CQL.toFilter("K1_ID="+parselNo));
collection = featureSource.getFeatures(filter);

iterator = collection.features();

try {
    while (iterator.hasNext()) {

        SimpleFeature feature = iterator.next();
        MultiPolygon poly = (MultiPolygon)
feature.getDefaultGeometry();
        Geometry boundary = poly.getBoundary();

        Coordinate[] coords = boundary.getCoordinates();

        sideLength = new double[coords.length - 1];

        for (int i = 0; i < sideLength.length; i++) {
            sideLength[i] = Point2D.distance(coords[i].x, coords[i].y,
            coords[i+1].x, coords[i+1].y);
        }

    }
} finally {
    if (iterator != null ){

        iterator.close();
    }

}

} catch (Exception ex) {
    ex.printStackTrace();
    System.exit(1);
}
return sideLength;
}
}

```


Ek 5. BinaKöşeKoordinatGetir Servis Kodu

```

package kadastro.servisler.tez.test;

import java.io.File;
import java.util.HashMap;
import java.util.Map;

import org.geotools.data.DataStore;
import org.geotools.data.DataStoreFinder;
import org.geotools.data.FeatureSource;
import org.geotools.factory.*;
import org.geotools.feature.FeatureCollection;
import org.geotools.feature.FeatureIterator;
import org.geotools.filter.text.cql2.CQL;
import org.opengis.feature.simple.SimpleFeature;
import org.opengis.feature.simple.SimpleFeatureType;
import org.opengis.filter.*;

import com.vividsolutions.jts.geom.*;

public class BinaKoordinatGetirServisi {

    public String BinaKoordinatGetir(String parselNo) {

        File file = new File("D:/MASTER/tezle ilgili/parsel/parsel.shp");
        File file2 = new File("D:/MASTER/tezle ilgili/bina/bina.shp");

        String koordinatlar= null;
        try{

            Map<String,Object> connect = new HashMap<String,Object>();
            connect.put("url", file.toURI().toURL());
            DataStore shapefile = DataStoreFinder.getDataStore(connect);
            String typeName = shapefile.getTypeNames()[0];

            FeatureSource<SimpleFeatureType, SimpleFeature> parcelfeature;
            FeatureCollection<SimpleFeatureType, SimpleFeature> parcelcoll;
            FeatureIterator<SimpleFeature> parceliterator;

            parcelfeature = shapefile.getFeatureSource(typeName);
            Filter filter1 = CQL.toFilter(("K1_ID="+parselNo));
            parcelcoll = parcelfeature.getFeatures(filter1);
            parceliterator = parcelcoll.features();

```

Ek 5'in devamı

```

try {

    SimpleFeature parcel= parceliterator.next();
    FilterFactory2 filterFactory = CommonFactoryFinder.getFilterFactory2(null);
    Geometry parcelGeom = (Geometry) parcel.getDefaultGeometry();

    Filter withinFilter = filterFactory.within(filterFactory.property("the_geom"),
    filterFactory.literal(parcelGeom));

    FeatureSource<SimpleFeatureType, SimpleFeature> buildingfeature;
    FeatureCollection<SimpleFeatureType, SimpleFeature> buildingsInParcel;
    FeatureIterator<SimpleFeature> buildingiterator;

        Map<String,Object> connect2 = new HashMap<String,Object>();
        connect2.put("url", file2.toURI().toURL());
        DataStore shapefile2 = DataStoreFinder.getDataStore(connect2);
        String typeName2 = shapefile2.getTypeNames()[0];
        buildingfeature=shapefile2.getFeatureSource(typeName2);

        buildingsInParcel = buildingfeature.getFeatures(withinFilter);

if (buildingsInParcel.isEmpty()) {
    koordinatlar="bina_yok";
} else {
    buildingiterator = buildingsInParcel.features();
    try {

        while (buildingiterator.hasNext()) {
            SimpleFeature building = (SimpleFeature) buildingiterator.next();
            Geometry geometry =
(Geometry)building.getDefaultGeometryProperty().getValue();
            Coordinate[] coords = geometry.getCoordinates();
            // Object attribute = building.getAttribute("BINA_");
            for( int i=0; i<coords.length; ++i) {
                koordinatlar=koordinatlar+coords[i].y+" "+coords[i].x+" ";
            }
        }
    } finally {if ( buildingiterator != null ){

        buildingiterator.close();
    }
}
}
}

```

Ek 5'in devamı

```
        }  
    }  
    finally {if( parceliterator != null ){  
        parceliterator.close();  
    }  
    } catch (Exception ex) {  
    ex.printStackTrace();  
    System.exit(1);  
    }  
    return koordinatlar;  
    }  
}
```

ÖZGEÇMİŞ

1984 yılında Trabzon'da doğdu. Orta ve lise öğrenimini Trabzon Yunus Emre Anadolu Lisesi'nde tamamladı. 2001 yılında, Karadeniz Teknik Üniversitesi Mühendislik Fakültesi Jeodezi ve Fotogrametri Mühendisliğinde lisans öğrenimine başladı. 2003–2004 eğitim öğretim yılında Erasmus bursuyla 1 sene Gent Üniversitesi'nde eğitim gördü. 2006 yılında lisans öğrenimini tamamlayarak aynı yıl, KTÜ Fen Bilimleri Enstitüsü Harita Mühendisliği Anabilim Kartografya bilim dalında yüksek lisans eğitimine başladı. 2006 yılında KTÜ Mühendislik Fakültesi'nde Araştırma Görevlisi olarak çalışmaya başladı. Halen bu görevine devam etmekte olup iyi derecede İngilizce bilmektedir.