

WEB SERVİSLERİNE DAYALI BİR KONUMSAL VERİ ALTYAPISINDA UYGULAMA GELİŞTİRME YAKLAŞIMLARI (APPLICATION DEVELOPMENT APPROACHES IN A WEB SERVICES BASED SPATIAL DATA INFRASTRUCTURE)

Halil AKINCI¹, Çetin CÖMERT²

¹Ondokuz Mayıs Üniversitesi Jeodezi ve Fotogrametri Mühendisliği Bölümü, Samsun
(hakinci@omu.edu.tr)

²Karadeniz Teknik Üniversitesi Jeodezi ve Fotogrametri Mühendisliği Bölümü, Trabzon
(ccomert@ktu.edu.tr)

ÖZET

Konumsal Veri Altyapıları (KVA), konumsal veri ile iş yapan bütün kesimler arasında “birlikte işlerliği” sağlayan ve vatandaşlar dahil ilgililere, gereksinim duydukları veri ve servislere anlık erişim ve kullanım olanağı sağlayan altyapılardır. KVA’lar, birlikte işlerlik altyapılarıdır. Birlikte işlerlik (interoperability), genel olarak yazılım ve donanım olarak farklı sistemlerin birbirleri ile “iletişim kurabilmesi” ya da “konuşabilmesi” şeklinde tanımlanabilir. Birlikte işlerliği gerçekleştirmek için geliştirilen en son yazılım mimarisi “Servis Yönelimli Mimari” (SyM) olarak adlandırılmaktadır. Web servisleri, SyM’yi gerçekleştirmenin en iyi yolu olarak kabul edilmektedir. SyM ve Web servisleri ortamında uygulama geliştirmenin yolu, servis düzenleme’dir. Servis düzenleme, kısaca, yeni uygulamalar yaratmak için mevcut Web servislerinin birleştirilmesi şeklinde tanımlanabilir. KVA’ların Web servisleri yaklaşımı ile gerçekleştirilebilmesi için dikkate alınması gereken, felsefe olarak birbirinden tamamen farklı iki servis mimarisi söz konusudur. Bunlar, Open Geospatial Consortium (OGC) tarafından geliştirilen OGC Web Servisleri mimarisi ve World Wide Web Consortium (W3C) tarafından geliştirilen W3C Web Servisleri mimarisidir. Söz konusu servis mimarilerinde, servis tanımlama, servis bulma ve servis düzenleme için farklı yöntemler ve standartlar kullanılmaktadır. Bu çalışmada, OGC ve W3C Web servisleri mimarileri açıklanacak ve her iki mimarideki servis düzenleme yaklaşımları anlatılacaktır.

Anahtar Kelimeler: Konumsal Veri Altyapısı (KVA), Servis Yönelimli Mimari (SyM), Web Servisleri, Servis Düzenleme

ABSTRACT

Spatial Data Infrastructures (SDIs) are infrastructures that allow interoperability between all parties that work with spatial data and allow ad-hoc access to required services and data for anyone who is interested as well as the citizens. SDIs are interoperability infrastructures for the spatial data. Interoperability can be defined as the ability by which different applications that use different languages or concepts can talk to each other. Service-Oriented Architecture (SOA) is the most popular and widespread software architecture which is designed to implement interoperability. Web services have been accepted as the best and the most popular way of implementing SOA. Service composition is the way of application development in SOA and Web services environment. Service composition can be defined as the process of combining existing Web services to create new applications. There are two service architectures philosophically different from each other to be taken into consideration to perform SDIs

with web services. These are OGC Web Services architecture developed by the Open Geospatial Consortium (OGC) and W3C Web Services architecture developed by the World Wide Web Consortium (W3C). In these architectures different methods and standards are used for service definition, service discovery and service composition. In this paper, OGC and W3C service architectures and their service composition approaches will be explained.

Key Words: Spatial Data Infrastructure (SDI), Service Oriented Architecture (SOA), Web Services, Service Composition

1. GİRİŞ

Kamu kurumları, özellikle doğal afetlerle, endüstriyel kazalarla, çevresel sorunlarla veya ulusal güvenliği tehdit eden terörist saldırılarla karşılaştıkları zaman, kamunun can ve mal güvenliğini korumak amacıyla, hızlı ve doğru kararlar verebilmek için konumsal veriye ihtiyaç duyarlar. Konumsal verinin elde edilmesi, zaman ve maliyet açısından oldukça zahmetlidir. Mevcut konumsal verinin ilgili tüm kullanıcılar tarafından paylaşımını sağlamak, böylece kurumlar tarafından mükerrer veri üretiminin önüne geçmek, veri toplama maliyetlerini düşürmek ve kurumların kaynak israfından kurtularak doğrudan hizmet üretimine geçmelerini sağlamak için “Konumsal Veri Altyapısı” (KVA) olarak adlandırılan birlikte işlerlik altyapılarına ihtiyaç duyulmaktadır. KVA’lar, konumsal veri ile iş yapan bütün kesimler arasında “birlikte işlerliği” sağlayan ve vatandaşlar dahil ilgililere, gereksinim duydukları veri ve servislere anlık erişim ve kullanım olanağı sağlayan altyapılardır. KVA’lar konumsal veri yönetimine yönelik birlikte işlerlik altyapılarıdır. Birlikte işlerlik (interoperability), genel olarak yazılım ve donanım olarak farklı sistemlerin birbirleri ile “iletişim kurabilmesi” ya da “konuşabilmesi” şeklinde tanımlanabilir. Farklı programlama dilleri kullanılarak geliştirilen, ağ üzerinde farklı yerlerde bulunan ve farklı platformlara sahip bilgisayarlar üzerinde koşan uygulamaların, birlikte işleyebilmelerine olanak sağlayan çeşitli sistemler ve yazılım mimarileri geliştirilmiştir. Şu an oldukça popüler ve yaygın olan yazılım mimarisi, Servis Yönelimli Mimari (Service Oriented Architecture), kısaca SyM, olarak adlandırılmaktadır.

SyM, uygulamaların son kullanıcılara servis olarak sunulduğu dağıtık sistemleri gerçekleştirmek için geliştirilen yeni bir yaklaşımdır (Colan, 2004). Web servisleri, SyM’yi gerçekleştirmenin en iyi ve şu anki en popüler yolu olarak kabul edilmektedir (McGovern, vd., 2003; Weerawarana vd., 2005). W3C (2002), bir Web servisini, İnternet tabanlı protokoller aracılığıyla XML tabanlı mesajları kullanarak diğer yazılım uygulamaları ile doğrudan etkileşimleri destekleyen, arayüzleri 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. Bir Web servisleri ortamı, servis sağlayıcılarının sahip oldukları Web servislerini bir katalog servisi aracılığıyla yayınladığı ve istemcilerin katalogdan servisleri bulup, uygulamalarını gerçekleştirmek için onları sağlayıcılardan istedikleri bir ortam olarak kavramsallaştırılabilir.

KVA’lar, kullanıcıların ham veri ile değil, coğrafi bilgi servisleri aracılığıyla ham verilerden üretilen bilgilerle ilgilendikleri varsayımı üzerine kurulmuştur (Bernard, vd., 2003). Bununla birlikte, KVA’ların ana bileşenleri olarak kabul edilen coğrafi bilgi servislerinin gerçekleştirilebilmesi için dikkate alınması gereken, felsefe olarak birbirinden tamamen farklı, iki servis mimarisi söz konusudur. Bunlar, OGC konsorsiyumu tarafından geliştirilen OGC Web Servisleri mimarisi ve W3C

konsorsiyumu tarafından geliştirilen W3C Web Servisleri mimarisidir. W3C tarafından geliştirilen Web servisleri mimarisi, Web Services Description Language (WSDL), Simple Object Access Protocol (SOAP) ve Universal Description, Discovery, and Integration (UDDI) standartları üzerine kurulmuş genel amaçlı bir mimaridir. W3C Web servisleri, günümüzde, elektronik ticaretten (e-commerce) tıbbi uygulamalara, içerik yönetiminden (content management) Coğrafi bilgi işleme ve coğrafi bilgi dağıtımına kadar birçok alanda yaygın olarak kullanılmaktadır. OGC Web servisleri mimarisi ise, Coğrafi Bilgi Sistemleri (CBS) arasındaki birlikte işlerlik problemlerini gidermek için geliştirilen ve sadece coğrafi bilgi işleme ve coğrafi bilgi dağıtımına yönelik olan özel amaçlı bir servis mimarisidir. Her iki mimaride de, servis tanımlama, servis bulma ve servis düzenleme için farklı yöntemler ve standartlar kullanılmaktadır. Bu çalışmada, OGC ve W3C Web servisleri mimarileri tanıtılacak ve her iki mimarideki servis tanımlama, servis bulma ve servis düzenleme yöntemleri açıklanacaktır.

2. SYM VE WEB SERVİSLERİ ORTAMINDA UYGULAMA GELİŞTİRME

SyM'de uygulama geliştirmenin yolu, "Servis Düzenleme" (SD) dir. SD bağlamında İngilizce literatürde, "servis composition", "servis chaining", "service orchestration" ve "service choreography" terimleri ile karşılaşılmaktadır. SD, bir kullanıcı tarafından gerçekleştirilmek istenen yeni bir uygulamanın, tek bir Web servisi tarafından gerçekleştirilememesi durumunda, mevcut Web servislerinin birleştirilerek uygulamanın gerçekleştirilmesi şeklinde tanımlanabilir. SD, kullanıcılara, farklı sağlayıcılar tarafından sunulan mevcut Web servislerini kullanarak uygulama geliştirme olanağı sağlamaktadır. SD sonucunda "birleşik servis" (composite service) adı verilen yeni bir Web servisi geliştirilir. Yeni düzenlenen serviste bir Web servisi gibi yayınlanabilir ve başka bir SD uygulamasında bileşen servis olarak kullanılabilir. Bir SD uygulaması, genel olarak üç temel işlemde oluşur. Bu işlemler sırasıyla, katalog servislerinde arama yapılarak uygulama gereksinimlerini karşılayan mevcut Web servislerinin bulunması, bulunan Web servislerinin girdi ve çıktı parametrelerinin karşılaştırılarak düzenlemede hangi servislerin hangi sırada kullanılacağına belirlenmesi ve son olarak servislerin belirlenen sırada yürütülmesidir. SD yöntemleri, bu işlemlerin "insan" veya "yazılım" tarafından yapılmasına bağlı olarak sınıflandırılmalıdır (Akıncı, 2006). SD, tasarım veya koşum zamanında, soyut ve somut düzeyde tanımlanabilir. Soyut ve somut SD, düzenlemenin insan ya da yazılım tarafından tanımlanmasına bağlı olarak, "otomatik olmayan", "yarı otomatik" ve "tam otomatik" olarak adlandırılan üç kategoriye ayrılabilir (Akıncı, 2006).

Otomatik olmayan yöntem, SD'nin insan tarafından gerçekleştirildiği yöntemdir. Bu yöntemde, mevcut Web servislerini kullanarak yeni bir uygulama geliştirmek isteyen bir kullanıcının, geliştireceği uygulama için ne tür Web servislerine ihtiyacı olduğunu ve Web servislerini nasıl düzenleyeceğini bilmesi gerekmektedir. Bu yöntemde kullanıcı ilk olarak, katalog servislerinde arama yaparak uygulama gereksinimlerini karşılayan Web servislerini bulmalıdır. Kullanıcı daha sonra, Web servislerinin girdi ve çıktı parametrelerini karşılaştırarak, eşleşen Web servislerini belirlemelidir. Kullanıcı, uygulama gereksinimlerini karşılayan Web servislerini belirledikten sonra, bir Web servisi geliştirme aracı kullanarak, uygulamayı gerçekleştiren yeni bir Web servisi yaratmalıdır. Yeni yaratılan Web servisi, "birleşik servis" (aggregate/composite service) olarak adlandırılır (ISO, 2001). Birleşik servis, uygulamayı gerçekleştirmek için, kullanıcı tarafından belirlenen "bileşen" servisleri kullanır. Birleşik servisin, bileşen servisleri hangi sırada çağıracağına, birleşik servisin gerçekleştirim kodunu yazan kullanıcı karar verir. Yarı-otomatik yöntem, SD'nin insan ve yazılım tarafından birlikte gerçekleştirildiği yöntemdir. Bu yöntemde, düzenlemede kullanılacak olan Web servisleri, insanlar tarafından katalog servislerinde arama yapılarak bulunur. Uygun Web servisleri

bulunduktan sonra, servis parametrelerinin karşılaştırılarak, eşleşen servislerin belirlenmesi bir yazılım (reasoner) tarafından otomatik olarak gerçekleştirilir. Eşleştirme anlamsal olarak yapılır. Bu nedenle Web servislerinin, WSDL gibi sözdizimsel bir dil yerine, DAML-S (DARPA Agent Markup Language for Services) veya OWL-S (Web Ontology Language for Services) gibi anlamsal bir dil kullanılarak tanımlanmış olması gerekmektedir. Tam otomatik SD, tüm SD işlem adımlarının bir yazılım tarafından gerçekleştirildiği yöntemdir. Tam otomatik SD uygulamalarını gerçekleştirmek için kullanılması gereken tekniklerin henüz tam olarak netleşmediği belirtilmektedir (Peer, 2005). Otomatik SD uygulamaları günümüzde, “Yapay Zeka Planlama” (Artificial Intelligence Planning) yöntemleri kullanılarak gerçekleştirilmektedir. Bu çalışmada, otomatik olmayan yöntemlere göre OGC ve W3C Web servisleri mimarilerinde bir SD uygulamasının nasıl gerçekleştirileceği üzerinde durulacaktır. SD yöntemleri ile ilgili ayrıntılı bilgiye (Akıncı, 2006)’dan erişilebilir.

3. W3C WEB SERVİSLERİ MİMARİSİ

Web alanındaki standartları geliştirmek için çalışan uluslararası bir birlik olan W3C Konsorsiyumu, World Wide Web (WWW)’in mucidi olan Tim Berners-Lee tarafından Ekim 1994 de kurulmuştur. W3C konsorsiyumunun temel amacı, Web’in uzun vadeli gelişimine olanak sağlayan protokoller ve yönergeler geliştirerek Web’in tam işlevsel duruma gelmesini sağlamak, en temel Web teknolojilerinin birbirleriyle uyumlu olmasını sağlamak ve Web’e erişmek için kullanılan her yazılım ve donanımın birlikte çalışabilmesine olanak sağlamaktır. W3C konsorsiyumu bu amacı, “Web birlikte işlerlik” (Web Interoperability) olarak adlandırmaktadır (W3C, 2008). W3C konsorsiyumu bu amaçla, kuruluşundan bugüne kadar, “W3C Önerisi” (W3C Recommendation) olarak adlandırılan 110 dan fazla standart geliştirmiştir. W3C’nin Web servisleri alanında geliştirdiği en temel standartların başında, WSDL ve SOAP gelmektedir. WSDL, Web servislerini tanımlamak için geliştirilen XML tabanlı bir dildir. SOAP ise, Web servisleri ile haberleşmeyi sağlayan XML tabanlı bir mesajlaşma protokolüdür.

W3C mimarisinde Web servisleri, WSDL dili kullanılarak tanımlanmaktadır. WSDL, Web servislerinin arayüzlerini tanımlayan, platformdan ve dilden bağımsız XML tabanlı bir dildir. WSDL, bir Web servisinin sahip olduğu operasyonları, operasyonların istek ve yanıt mesajlarını, mesajlar aracılığıyla iletilecek olan verileri (operasyon parametrelerini) ve tiplerini, Web servisini çağırmak için kullanılacak olan uygulama protokolünü ve Web servisinin adresini tanımlar. Bir WSDL dokümanı, types, interfaces, binding ve service olarak adlandırılan dört elementten oluşur. Types elementi, istemci ile Web servisi arasında mübadele edilen mesajları tanımlar. Interface elementi, Web servisinin gerçekleştirdiği operasyonları tanımlar. Binding elementi, interface tarafından tanımlanan operasyonlar ve mesajlar için mesaj formatını ve protokol detaylarını tanımlar (Nagappan vd., 2003). Service elementi ise, servise nereden erişileceğini tanımlar ve Web servisinin internet adresini içerir.

W3C Web servisleri mimarisinde, bir kullanıcının aradığı servisleri bulmasını sağlayan bileşen, katalog servisleridir. Katalog servisleri, SyM’nin kilit bileşeni durumundadır. Çünkü SyM’de servis sağlayıcılar Web servislerini katalog servisleri üzerinden kullanıma sunar, istemciler de aradıkları özellikteki Web servislerini bu sayede tespit ederler. Diğer bir anlatımla katalog servisi, sağlayıcıların Web servislerini katalog servisine kaydetmelerine ve istemcilerin de istenen özellikleri sağlayan Web servislerini aramalarına olanak tanır. Burada söz konusu olan iki işlem, sırasıyla, SyM’nin “yayınla” (publish) ve “bul” (find) işlemleridir. W3C Web servisleri mimarisinde, servisleri kataloglamak için OASIS (Organization for the Advancement of Structured

Information Standards) tarafından standart olarak kabul edilen UDDI ve ebXML Katalog Servisi (ebXML KS) kullanılmaktadır. ebXML KS, her türlü bilgi kaynağını tanımlamaya olanak sağlayan nesne yönelimli bir bilgi modeline sahiptir. UDDI ise, sadece Web servislerini ve servis sağlayıcılarını tanımlamaya olanak sağlayan bir bilgi modeline sahiptir. UDDI ve ebXML KS ile ilgili detaylı bilgiye (Akıncı, 2006) dan erişilebilir.

W3C Web servisleri mimarisinde, bir SD uygulaması gerçekleştirmek için izlenebilecek iki yöntem vardır. SD literatüründe “hard-coded” yaklaşım olarak adlandırılan birinci yöntemde, SD’yi gerçekleştiren birleşik Web servisi bir programlama dili kullanılarak geliştirilir. Düzenlemede kullanılacak olan bileşen Web servisleri, katalog servislerinde arama yapılarak bulunur. İstemcilerden gelen istek mesajlarının alınması, servis parametrelerinin eşleştirilmesi (bir servisin döndürdüğü değer bir sonraki servise girdi olarak atanması), servislerin belirlenen sırada yürütülmesi ve yanıt mesajlarının üretilerek istemcilere gönderilmesi gibi işlemler, birleşik servisi geliştiren kullanıcı tarafından birleşik servisin gerçekleştirim koduna yazılır. Geliştirilen yeni Web servisi, bir Web servisleri yayımlama aracı kullanılarak istemcilerin kullanımına sunulur. Bu yöntemin en büyük dezavantajı, bileşen servislerin arayüzlerinin değişmesi durumunda, birleşik servisin gerçekleştirim kodunun yeniden yazılması gerekmektedir.

Web servisleri ile ilgili teknolojiler geliştiren yazılım firmaları, SD’nin kolay ve hızlı bir şekilde gerçekleştirilmesine olanak sağlayan çeşitli SD dilleri geliştirmiştir. Günümüzde en yaygın kullanılan SD dili, Web Services Business Process Execution Language (WSBPEL)’dir. BPEL, XML ve WSDL standartları üzerine kurulmuş bir dildir. Bir BPEL işlemi (BPEL process), mevcut Web servislerini kullanarak yeni bir uygulama geliştirmek için, BPEL şemasına uygun olarak geliştirilen bir XML dokümanıdır. Bir BPEL işlemi, düzenlemede kullanılan Web servislerini çağırmak için, bu servislerin WSDL dokümanlarında tanımlanan operasyonları ve mesajları kullanır. BPEL işlemleri, bir WSDL dokümanı ile tanımlanır ve bir Web servisi olarak sunulabilir. Bir BPEL işlemi, SD’yi gerçekleştirmek için çeşitli aktiviteler kullanır. Bu aktiviteler, “basit aktiviteler” ve “yapısal aktiviteler” olmak üzere ikiye ayrılır (OASIS, 2007). Basit aktiviteler, istemcilerden gelen istek mesajlarını almak, değişkenlere değer atamak, servislerin operasyonlarını çağırmak, istemciye bir yanıt mesajı göndermek, koşum anında oluşan hataları göstermek, BPEL işlemini belirli bir süre bekletmek ve sonlandırmak gibi temel görevleri yerine getirirler. Yapısal aktiviteler ise, basit aktiviteleri içerirler ve bu aktivitelerin hangi sırada yürütüleceklerini belirtirler. Bir BPEL işleminin içerdiği XML kodlarının insanlar tarafından elle yazılması oldukça zordur ve hata eğilimlidir. Bu nedenle birçok yazılım firması, BPEL işlemlerinin kolay bir şekilde geliştirilmesine olanak sağlayan araçlar geliştirmiştir. BPEL tasarımcısı (BPEL designer) olarak adlandırılan bu araçlar, kullanıcıların BPEL işlemlerini, BPEL aktivitelerini temsil eden nesnelere kullanarak, bir iş akış şeması oluşturur gibi grafik olarak oluşturmalarına olanak sağlamakta ve BPEL işlemlerini bu grafik gösterimlerden otomatik olarak üretmektedirler. BPEL işlemleri, BPEL motoru (BPEL engine) olarak adlandırılan yazılımlar tarafından yürütülürler. Bu nedenle, BPEL tasarımcısında üretilen BPEL işlemlerinin bir BPEL motoruna yayımlanması gerekmektedir. SD’nin BPEL dili kullanılarak gerçekleştirilmesinin en önemli avantajı, düzenlemede kullanılan Web servislerinin arayüzlerinin değişmesi durumunda, BPEL işleminin güncellenmesinin veya yeniden üretilmesinin kolay olmasıdır.

4. OGC WEB SERVİSLERİ MİMARİSİ

1994 yılında kurulan OGC, CBS şirketlerinden, kamu kurumlarından ve üniversitelerden oluşan, yaklaşık 400 üyeye sahip, kar amacı gütmeyen uluslararası bir

endüstri birliğidir. Birliğin amacı, mevcut CBS sistemleri arasındaki birlikte işlerlik problemlerini belirlemek ve bilgi teknolojilerindeki gelişmeleri yakından izleyerek bu problemleri giderecek, herkesin kullanımına açık arayüz belirtileri geliştirmektir. OGC, bu amaç doğrultusunda çeşitli birlikte işlerlik girişimleri başlatmış ve bu girişimlerin sonucunda, Web Map Service (WMS), Web Feature Service (WFS), Web Coverage Service (WCS) ve Web Processing Service (WPS) gibi çeşitli servisler geliştirmiştir.

OGC, Web servislerinin yapısını şekillendiren kapsamlı bir servis modeline sahiptir (Doyle ve Reed, 2001). Bu modele göre her OGC Web servisi, operasyonlarından biri GetCapabilities olan bir arayüze sahip olmak zorundadır. GetCapabilities operasyonu, istemcilere, bir OGC Web servisinin desteklediği operasyonları ve sunduğu konumsal verileri tanımlayan ve "servis metaverisi" (service metadata) olarak adlandırılan bir XML dokümanı döndürür. Servis metaveri dokümanı, servisin "yetenekler dokümanı" (capabilities document) olarak ta adlandırılmaktadır. Servis metaveri dokümanları, "ServiceIdentification", "ServiceProvider", "OperationsMetadata" ve "Contents" olarak adlandırılan dört bölümden oluşur (OGC, 2005). ServiceIdentification bölümü, OGC Web servisi ile ilgili temel metaverileri içerir. ServiceProvider bölümü, servisi sunan organizasyonla ilgili bilgiler içerir. OperationsMetadata bölümü, servisin gerçekleştirdiği operasyonları ve parametrelerini tanımlayan bölümdür. Contents ise, servis tarafından sunulan konumsal verilerle ilgili metaverilerin tanımlandığı bölümdür.

OGC Web servisleri mimarisi, kavramsal olarak SyM'yi gerçekleştirmektedir. OGC Web servisleri mimarisinde, bir kullanıcının aradığı servisleri bulmasını sağlayan bileşen, CSW (Catalogue Service Web) katalog servisleridir. OGC Web servisleri, servis sağlayıcıları tarafından CSW katalog servislerine kayıt edilirler. Kullanıcılar, katalog servislerinde arama yaparak ihtiyaç duydukları Web servislerini bulabilirler.

OGC Web servisleri, konumsal veri sunmak için geliştirilmiş servislerdir ve mevcut işleyişleri ile uygulamaların insanlar tarafından gerçekleştirilmesine olanak sağlayan (insan-odaklı) bir modele sahiptir. Bu nedenle, OGC Web servislerini kullanarak bir SD uygulaması gerçekleştirmek oldukça zor ve zahmetli bir işittir.

OGC mimarisinde SD'yi gerçekleştirmenin dört yolu vardır. Bunlardan birincisi, OGC Web servislerini BPEL işlemleri içerisinde kullanarak SD'yi gerçekleştirmektir. Bu yöntem, OGC Web servislerinin WSDL dili kullanılarak tanımlanmış olmasını gerektirmektedir. OGC, Web servislerinin WSDL tanımlarını üretmek için çeşitli çalışmalar yapmış ve bu çalışmalar sonucunda OGC Web servislerini tanımlayan WSDL dokümanlarını geliştirmiştir (OGC, 2003; OGC, 2004). OGC Web servislerinin WSDL tanımlarını üretmek, bu servisleri bir BPEL işleminde kullanarak SD'yi gerçekleştirmek için tek başına yeterli değildir. Çünkü OGC Web servislerinden konumsal verileri elde etmek için kullanılan operasyonlar, parametre olarak bir sorgu ifadesi içermekte ve bu sorgu ifadeleri kullanıcılar tarafından yazılmaktadır. BPEL belirtimi, sorgu ifadelerinin BPEL işlemleri içerisinde oluşturulmasına olanak sağlayan aktivitelere sahip değildir. Bu nedenle, OGC Web servisleri ile SD'yi gerçekleştirebilmek için, sorgu ifadelerinin BPEL işlemleri içerisinde oluşturulmasına olanak sağlayan araçlara ihtiyaç vardır. IBM ve BEA yazılım firmaları tarafından geliştirilen BPEL4J (BPEL for Java) (Blow vd., 2004) dili, BPEL ve Java programlama dillerinin bir BPEL işleminde birlikte kullanılmasına olanak sağlamaktadır. BPEL4J, "Java snippets" olarak adlandırılan ve döngü koşulları, dallanma koşulları, değişkenlerin hazırlanması ve servis mesajlarının hazırlanması gibi işlemleri yerine getiren küçük Java kodlarının, BPEL işlemleri içerisinde kullanılmasına olanak sağlamaktadır. Söz konusu Java kodları, BPEL işlemleri içerisinde, OGC Web servislerine gönderilecek olan sorgu ifadelerini

oluşturulmak için kullanılabilir. Bu nedenle, BPEL4J kullanılarak OGC Web servisleri ile SD uygulamaları geliştirilebilir.

OGC Web servisleri ile SD'yi gerçekleştirmenin ikinci yolu, bir SD uygulamasında OGC Web servislerini düzenlemenin son servisi olarak kullanmaktır. Bu yöntemde, OGC ve W3C Web servisleri bir arada kullanılmakta ve kullanıcıların ihtiyaç duyduğu haritalar düzenlemenin son servisi olarak kullanılan OGC Web servisleri tarafından üretilmektedir. Lemmens vd. (2006), bu yöntemi kullanarak bir şehirdeki potansiyel tehlikeli alanları gösteren bir risk haritası üreten örnek bir uygulama geliştirmiştir. OGC Web servislerini kullanarak SD'yi gerçekleştirmenin üçüncü yolu, düzenlemede kullanılacak olan her bir OGC Web servisi için bir W3C Web servisi geliştirmek ve SD'yi W3C servislerini kullanarak BPEL dilinde tanımlamaktır. Bu yöntemde, OGC Web servislerinden istekte bulunanlar, W3C Web servisleridir. W3C servisleri, istemciler ile SOAP protokolünü kullanarak haberleşirler. İstemcilerden SOAP protokolünü kullanarak aldıkları istek mesajlarını, OGC isteklerine dönüştürürler ve OGC Web servisleri ile HTTP protokolünü kullanarak haberleşirler. Bu yöntemde kullanılan W3C Web servisleri, "Wrapper Servis" olarak adlandırılırlar. OGC Web servisleri ile SD'yi gerçekleştirmenin son yolu ise, bir OGC WPS servisi kullanmaktır. WPS, Web üzerinden CBS konumsal analiz fonksiyonlarını gerçekleştirmek için geliştirilmiştir. Kullanıcılar bir WPS servisini kullanarak, tampon bölge oluşturmak veya poligon-poligon bindirmesi yapmak gibi temel CBS analiz fonksiyonlarını Web üzerinden gerçekleştirebilirler. WPS servisi, analizlerde kullanacağı konumsal verileri, WMS, WFS veya WCS gibi OGC Web servislerinden elde etmektedir. Stollberg ve Zipf (2007), bu yöntemi kullanarak, bir yerleşim gölgesinde bulunan bir bombanın tehdidi altındaki binaları belirleyen ve bu binalardaki insanların tehlikeli bölgeden güvenli bir şekilde tahliye edilmesini sağlayan örnek bir SD uygulaması geliştirmiştir.

5. SONUÇ

KVA'lar konumsal veri yönetimine yönelik birlikte işlerlik altyapılarıdır. Birlikte işlerliği gerçekleştirmek için geliştirilen en son kuşak Web teknolojisi, SyM ve Web Servisleridir. Bununla birlikte, KVA'ların Web servisleri teknolojileri ile gerçekleştirilebilmeleri Dünya genelinde henüz başarısızdır. Çünkü böyle bir gerçekleştirmenin, teknik birlikte işlerlik altyapısını tanımlayan, üzerinde anlaşma sağlanmış bir çerçeve mevcut değildir. Bu durum, Web servislerine dayalı bir KVA gerçekleştirmek isteyenlerin işini, gerek teknik gerçekleştirmeciler ve gerekse KVA'ları kurmak ve yaşatmaktan sorumlu karar vericiler bazında çok zor kılmaktadır. KVA'lar günümüzde, "insan-odaklı Web" niteliğine uyan bir teknoloji olan portal teknolojisi kullanılarak gerçekleştirilmektedir. Portal tabanlı KVA'lar, OGC Web servisleri mimarisi üzerine kurulmuştur. OGC'nin CSW belirtimini gerçekleştiren merkezi bir metaveri kataloguna sahip olan mevcut KVA gerçekleştirmelerinde kullanıcılar, portal katalogunda arama yaparak geliştirmek istedikleri uygulamaların gereksinimlerini karşılayan konumsal verileri bulmakta ve bu verileri bir harita görüntüleme aracında görüntülemektedirler. Harita görüntüleme aracı, uygulama geliştirmek için ihtiyaç duyulan CBS analiz fonksiyonlarını sunmamaktadır. Bu nedenle, portal tabanlı KVA gerçekleştirmelerinde insan kullanıcılar, uygulamalarını başka bir sistemde geliştirmek durumundadırlar. Bunda, OGC Web servisleri mimarisinde uygulama geliştirmenin oldukça zor ve zahmetli olmasının da etkisi vardır. Bu nedenle, portal tabanlı KVA'ların işlevlerini, sadece, "bul ve görüntüle" olarak kavramsallaştırabiliriz. Web servisleri teknolojileri henüz olgunlaşmakta olan teknolojiler olarak kabul edilse de, nispeten durağan konuma gelmiş bileşenler söz konusudur. Örneğin, WSDL ve SOAP yaygın olarak kullanılmaktadır. Aynı şekilde WSBPEL SD için kullanılabilecek durumdadır ve BPEL ile uygulama geliştirmek oldukça kolaydır. Bu

bakımdan, W3C Web servisleri ile mevcut KVA market gerçekleştirmelerinden daha iyisinin yapılabileceği ortadadır.

KAYNAKLAR

- Akıncı, H.**, (2006). Konumsal Veri Altyapılarının Web Servisleri ile Gerçekleştirilmesi: Mevcut Durum Analizi ve Gelecek Yönelimlerinin Belirlenmesi, *Doktora Tezi*, KTÜ Fen Bilimleri Enstitüsü, Trabzon.
- Bernard, L., Einspanier, U., Haubrock, S., Hübner, S., Kuhn, W., Lessing, R., Lutz, M. ve Visser, U.**, (2003). Ontologies for Intelligent Search and Semantic Translation in Spatial Data Infrastructures, *Photogrammetrie-Fernerkundung-Geoinformation*, 451-462.
- Blow, M., Goland, Y., Kloppmann, M., Leymann, F., Pfau, G., Roller, D. ve Rowley, M.**, (2004). BPELJ: BPEL for Java Technology, A Joint White Paper by BEA and IBM, <http://www-128.ibm.com/developerworks/library/specification/ws-bpelj/>, (2005.01.09)
- Colan, M.**, (2004). Service-Oriented Architecture expands the vision of Web services, Part 1: Characteristics of Service-Oriented Architecture, <http://www-128.ibm.com/developerworks/webservices/library/ws-soaintro.html> (2004.11.13)
- Doyle, A. ve Reed, C.**, (2001). Introduction to OGC Web Services, OGC Interoperability Program White Paper, http://ip.opengis.org/ows/010526_OWWhitepaper.doc, (2004.11.09)
- ISO**, (2001). Geographic information — Services, ISO/DIS 19119, ISO TC 211/WG 4.
- Lemmens, R., Granell, C., Wytzisk, A., de By, R., Gould, M. ve Oosterom, P. V.**, (2006). Semantic and syntactic service description at work in geo-service chaining, 9th AGILE Conference on Geographic Information Science, Visegrad, Hungary.
- McGovern, J., Tyagi, S., Stevens, M. ve Mathew S.**, (2003). Java Web Services Architecture, Morgan Kaufmann, San Francisco, USA.
- Nagappan, R., Skoczylas, R. ve Sriganesh, R. P.**, (2003). Developing Java Web Services, Wiley Publishing Inc., Indiana, USA.
- Newcomer, E.**, (2002). Understanding Web Services XML, WSDL, SOAP and UDDI, Addison-Wesley, Indianapolis, USA.
- OASIS**, (2005a). ebXML Registry Services and Protocols, Version 3, OASIS Standard.
- OASIS**, (2005b). ebXML Registry Information Model, Version 3.0, OASIS Standard.
- OASIS**, (2007). Web Services Business Process Execution Language, OASIS Standard
- OGC**, (2003). OWS 1.2 SOAP Experiment Report, OGC 03-014, Version: 0.8, OGC Discussion Paper.
- OGC**, (2004). OWS 2 Common Architecture: WSDL SOAP UDDI, OGC 04-060r1, Version: 1.0.0, OGC Discussion Paper.
- OGC**, (2005). OGC Web Services Common Specification, Version 1.0.0, OGC 05-008, OGC Implementation Specification.
- Peer, J.**, (2005). Web Service Composition as AI Planning - a Survey, Technical Report, University of St.Gallen, Switzerland.
- Stollberg, B. ve Zipf, A.**, (2007). OGC Web Processing Service Interface for Web Service Orchestration, Aggregating Geo-processing Services in a Bomb Threat Scenario, *Lecture Notes in Computer Science*, Volume 4857/2007, 239-251.
- Weerawarana, S., Curbera, F., Leymann, F., Storey, T. ve Ferguson, D. F.**, (2005). Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More, Prentice Hall Ptr, USA.
- W3C**, (2002). Web Services Description Requirements, W3C Working Draft, <http://www.w3.org/TR/ws-desc-reqs/>, (2005.11.11).
- W3C**, (2008). About the World Wide Web Consortium, <http://www.w3.org/Consortium/>, (2009.02.03)