



T.C.

SELÇUK ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

HARİTA MASHUP UYGULAMALARIYLA ZAMANA VE MEKÂNA BAĞLI GÖRSEL ANALİTİK ORTAMI OLUŞTURULMASI: KONYA KAPALI HAVZASI ÖRNEĞİ

Osman Sami KIRTILOĞLU

DOKTORA TEZİ

Harita Mühendisliği Anabilim Dalını

Nisan-2016
KONYA
Her Hakkı Saklıdır

ÖZET

DOKTORA TEZİ

HARİTA MASHUP UYGULAMALARIYLA ZAMANA VE MEKÂNA BAĞLI GÖRSEL ANALİTİK ORTAMI OLUŞTURULMASI: KONYA KAPALI HAVZASI ÖRNEĞİ

Osman Sami KIRTILOĞLU

Selçuk Üniversitesi Fen Bilimleri Enstitüsü
Harita Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. İ. Öztuğ BİLDİRİCİ

2016, 119 Sayfa

Jüri
Prof. Dr. İ. Öztuğ BİLDİRİCİ
Prof. Dr. N. Necla ULUĞTEKİN
Doç. Dr. Aydın ÜSTÜN
Doç. Dr. R. Alpay ABBAK
Doç. Dr. Ekrem TUŞAT

Bu çalışmada, web ve web'in sunduğu yeni teknolojiler, Konya Kapalı Havzası (KKH) sınırları içerisinde zemin çökmelerinin ve nedenlerinin araştırılması için etkili ve etkileşimli görsel ara yüzlerin oluşturulmasında, temel araç olarak kullanılmıştır. Çalışma kapsamında zemin çökmelerini tetikleyebilecek farklı parametreler için çoklu kaynaklardan elde edilen veriler analiz edilmiştir. Sonuçların bir Coğrafi Bilgi Sistemi (CBS) altyapısında, ücretsiz web tabanlı haritalama ve diğer amaçlara uygun servisler kullanılarak, web üzerinden dağıtımının yapılabilmesi için, bir harita mashup uygulaması oluşturulmuştur. Uygulamayla, mevcut verilerin ve bu veriler arasındaki ilişkilerin keşfedilmesinde etkili ve anlaşılabilir görsel yöntem ve araçların kullanılması amaçlanmıştır. Bu amaç doğrultusunda bilgi yönetimi, istatistiksel analizler, karar bilimleri ve daha birçok alanı bütüncül bir yaklaşımla birleştiren görsel analitik yaklaşımı ve bu yaklaşımın bir alt dalı olan mekânsal görsel analitik yaklaşımlarına odaklanılmıştır. Temel amaç, mashup yaklaşımına dayanan ve CBS fonksiyonelliklerine sahip bir mekânsal görsel analitik ortamının değerlendirilmesi, tasarlanması ve geliştirilmesidir.

Çalışma bölgesi, KKH, yaklaşık 62.000 km^2 alana sahip, Anadolu Yarımadası'nın iç kesiminde bulunan, Konya, Karaman, Niğde ve Aksaray illerini de içine alan Türkiye'nin en büyük kapalı havzasıdır. Havza sınırları içerisinde farklı kaynaklardan elde edilen veriler GPS (Global Positioning System)/GNSS (Global Navigation Satellite System) gözlemleri, inklinometrik ölçümler, yeraltı su seviyesi ölçümleri, meteorolojik yağış verileri, radar uydusu verileri (InSAR) ve normalleştirilmiş fark bitki endeksi (NDVI) verilerini içermektedir. Bu verilerle gerçekleştirilen analizlerin sonuçlarının, web üzerinden dağıtımının yapılabilmesi için Google Map API V3 (Application Programming Interface), Google Fusion Tables API, AJAX, Google Drive ve High Charts API servisleri kullanılmaktadır. Uygulama aynı zamanda esneklik ve daha fazla etkileşim için HTML (HyperText Markup Language), CSS (Cascading Style Sheet) ve JavaScript gibi web istemci teknolojilerini de kullanmaktadır. Bu teknolojiler kullanılarak web-tabanlı harita servislerinin hali hazırda sunmadığı, dış verinin katman olarak eklenmesi durumunda katman gruplandırma ve eklenen katmanların dinamik lejant alanlarının oluşturulması problemine bir çözüm geliştirilmiştir. Ayrıca genel amaçları konum paylaşmak, konum tabanlı aramalar ve reklam yayınları olan web harita servisleri ile gelişmiş mashup uygulamaları oluşturulması konusunda, mevcut servislerin henüz sunmadığı izoplet veri katmanlarının oluşturulması

problemine yeni bir yaklaşımla çözüm sunulmaya çalışılmıştır. “Point in Polygon” ve “Multikuadrik Radyal Bazlı Fonksiyon” yaklaşımının kullanıldığı çözüm ile kullanıcılar, istatistiksel yüzeyleri oluşturabilmekte ve web-tabanlı harita servislerine katman olarak ekleyerek izoplet haritalar oluşturabilmektedir. Harita üzerine eklenen katmanların zamana bağlı olarak değişimlerinin izlenmesi için “Swipe Mode” adı verilen yeni bir ara yüz tasarlanmıştır. Bu ara yüz ile iki farklı katman eş zamanlı olarak harita üzerine eklenebilmekte ve geliştirilen ekran süpürme aracıyla katmanlar arasındaki değişimlerin izlenmesi sağlanmaktadır. Zamana bağlı değişimlerin izlenmesi için geliştirilen bir diğer yaklaşım ise zaman kaydırma çubuğu aracıdır. Bu araç, harita üzerine eklenecek katmanların zamana bağlı olarak çağırılmasını sağlamaktadır.

Geliştirilen tüm görselleştirme bileşenleri herhangi bir web tarayıcısında çalışabilmekte ve kullanıcı dostu bir ara yüz sağlamaktadır. Hazırlanan arayüzle, uygun yöntem ve teknikler değerlendirilerek kullanılabilirlik, erişim kolaylığı, kısıtlamalar ve gizlilik politikası yönünden detaylı olarak incelenmiştir. Her ne kadar CBS işlevsellikleri sunan mashup uygulamalarının geliştirilmesi için uygun teknoloji ortamı oluşsa da, uzun zamandır tam CBS çözümleri sunan güçlü masaüstü platformlarla arasında işlevsellik bakımından daima bir boşluk bulunmaktadır. Tezde elde edilen katkıların bir araya getirilmesi ve hayata geçirilmesi ile web-tabanlı harita mashup uygulamaları ve güçlü masaüstü CBS uygulamaları arasındaki köprünün inşası için bir yapı taşı daha oluşturulmuştur.

Anahtar Kelimeler: Coğrafi Bilgi Sistemi, Google Maps API, Kartografya, Konya Kapalı Havzası, Mash-Up, Mekânsal Görsel Analitik, Web Kartografya, Web haritalama servisleri,

ABSTRACT

Ph.D THESIS

DEVELOPING A GEO-TEMPORAL VISUAL ANALYTICS ENVIRONMENT WITH MAP MASHUPS: A CASE STUDY ON KONYA CLOSED BASIN

Osman Sami KIRTILOĞLU

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF SELÇUK UNIVERSITY
DOCTOR OF PHILOSOPHY
IN GEOMATICS ENGINEERING

Advisor: Prof.Dr. İ.Öztuğ BİLDİRİCİ

2016, 119 Pages

Jury
Prof.Dr. İ.Öztuğ BİLDİRİCİ
Prof. Dr. N. Necla ULUĞTEKİN
Assoc. Prof. Dr. Aydın ÜSTÜN
Assoc. Prof. Dr. R. Alpay ABBAK
Assoc. Prof. Dr. Ekrem TUŞAT

In this study, we use the web and its emerging technologies as the basis of creating an efficient and interactive visual interface for investigating the results of vertical displacements and its causes in Konya Closed Basin (KCB). In the scope of the work the data obtained from multiple sources that would trigger the ground deformation was analyzed. A map mashup application was created according to geographical information system (GIS) infrastructures for disseminating the analysis results by using free web-based mapping services. Another focus point of the thesis is to use effective and understandable visual tools and methods for discovering available data and their relations by the application that will be created. In accordance to this purpose we focused on the visual analytics concept which integrates data management, statical analysis, decision sciences and many others in a holistic way. Geovisual analytics is a sub-domain of visual analytics and focuses on the geospatial knowledge construction and dissemination from massive geospatial data. The main objective is to assess, design and develop a GIS-like geovisual analytics environment, based on Mashup architecture.

The study area KCB, is the Turkey's largest closed basin, located at the inner part of the Anatolian peninsula, covering an area of approximately 62,000 km^2, and includes the provinces of Konya, Karaman, Nigde and Aksaray. The multi-sourced data in closed basin include GPS (Global Positioning System)/GNSS (Global Navigation Satellite System) observations, inclinometric and ground water level measurements, meteorological precipitation data, radar satellite data (InSAR) and Normalized Difference Vegetation Index (NDVI) data. The application uses Google Maps API V3, Google Fusion Tables API, Google Drive, AJAX and High Charts API to disseminate the obtained results through the Web. The application also utilized Web client technologies such as HTML (HyperText Markup Language), CSS (Cascading Style Sheet), and JavaScript for handling the response events and providing flexibility and more interactivity. A solution has been developed for layer classification of the added layers on the web map and creating dynamic legend windows of these layers when adding an external data layer on the map. In addition, the problem of creating isopleth maps with web mapping services has been discussed

and a new tool which uses the “Point in Polygon Test” and “Multiquadric Radial Based Function” approaches has been generated as a solution for this problem. By using this tool, web users can create their own ispoleth data layers with their own statistical data and add this data layer to any web mapping service base map. To compare different layers in a single interface we created a new visual interface named “Swipe Mode”. By this tool, users can add two different layers in a single interface as right and left layers and by using the drag bar they can swipe the screen to left or right to compare layers. Another tool to provide geo-temporal display of the layers is time-slider tool. By sliding the time bar to left and right, users can change the map display according to selected time.

All visualization components used in the development of mashup applications can run on any web browser and provides a user-friendly interface. In developing process of the application, previous studies have been discussed in detail; appropriate methods and techniques reviewed and have been evaluated in terms of usability, accessibility, limitations and privacy policy. Although there is an appropriate environment exists today for developing mashup applications that provide GIS functionality, there is always a gap between powerful desktop GIS applications and web-based map mashups in terms of functionality. Bringing together all these contributions, one more building-block can be added to bridging the gap between desktop GIS applications and web-based map mashup applications.

Keywords: Cartography, Geographical Information System, Google Maps API, Konya Closed Basin, Mash-Up, Web Cartography, Web Mapping, Web mapping services

ÖNSÖZ

Doktora derecesine sahip olmak danışman, tez izleme komitesi üyeleri ve sizi seven insanların desteği olmadan aşılamayacak uzun, sabır gerektiren ve çok çalışma isteyen bir yoldur. Burada doktora tezim süresince beni destekleyen ve çalışmalarımda yardımcı olan önemli insanlara teşekkür etmek istiyorum. Bu konuda öncelikle tez danışmanım sayın Prof. Dr. İbrahim Öztuğ BİLDİRİCİ'ye teşekkür ediyorum. Web kartografya ve web geliştirme alanında sahip olduğu bilgilerle önemli yönlendirmelerde bulunmuş, çalışmalarımın şekillenmesinde ilham kaynağı olmuştur. Ayrıca tez süresi boyunca göstermiş olduğu çözüm odaklı ve pozitif yaklaşımlar, çalışma alanıma duyduğu ilgi, büyük motivasyon kaynağı olmuştur.

Bu tez her ne kadar benim çalışmalarım ve gayretlerimle ortaya çıkmış gibi görünse de büyük ölçüde ortak bir çalışmanın ürünüdür. Tez izleme komitesi üyelerinden İstanbul Teknik Üniversitesi Geomatik Bölümü öğretim üyesi sayın hocam Prof. Dr. Nesibe Necla ULUĞTEKİN, tezin tüm aşamalarında yanımda olmuş, desteğini esirgememiş, uzun yollar kat etmeyi göze alarak tez izleme komitesi üyeliğini kabul etmiş ve önemli yönlendirmeleriyle tezin şekillendirilmesinde öncü olmuştur. Kendisi ile çalışmak benim için hem zevk hem de gurur kaynağıdır, bunun için kendisine şükranlarımı sunarım.

Bu tezde gerçekleştirilen durum çalışmasında, Türkiye Bilimsel ve Teknolojik Araştırma Kurumu (TÜBİTAK) tarafından desteklenen 110Y121 numaralı "Konya Kapalı Havzasında Zemin Çökmelerinin Jeodezik Yöntemlerle İzlenmesi ve Nedenlerinin Araştırılması" projesinden elde edilen veriler kullanılmıştır. Bu projenin yürütücüsü ve aynı zamanda tez izleme komitesi üyesi sayın hocam Doç. Dr. Aydın ÜSTÜN'e de teşekkürü borç bilirim. Tezde kullanılan tüm verilere ulaşım, durum çalışması test aşamasında sağladığı teknik ve teorik destek ve teze yaptığı katkılar için minnettarım. Ayrıca çalışmamda kullandığım GPS istasyonlarının veri analizleri ile InSAR görüntü çiftlerinin oluşturulmasında, dostlarım ve çalışma arkadaşlarım Arş. Gör. Sefa YALVAÇ ve Yüksek Müh. Fatma CANASLAN ÇOMUT' a teşekkür ederim.

Sonuncu ama bir o kadar da önemli olarak en büyük teşekkürü aileme sunuyorum. Uzun çalışma saatleri, uykusuz geceler ve sosyal hayatımızdan fedakârlık konularında beni sabırla destekleyen ve desteğini bir an bile benden esirgemeyen sevgili eşim Yüksek Müh. Elif KIRTILOĞLU' na teşekkür ederim. Bana her zaman inanan ve başarımı görmek için sabırsızlanan anne ve babama, beni çok seven ablama ve bu süreçte her zaman desteklerini sunan kayınpederim ve kayınvalideme ayrıca teşekkür ederim.

Burada adını sayamadığım, ancak az ya da çok, tez çalışmamda katkıda bulunan Selçuk Üniversitesi Harita Mühendisliği Bölümü akademik personeline de çok teşekkür ederim

Osman Sami KIRTILOĞLU
KONYA-2016

İÇİNDEKİLER

ÖZET ..... **iv**

ABSTRACT ..... **vi**

ÖNSÖZ ..... **viii**

SİMGELER VE KISALTMALAR ..... **xi**

1 GİRİŞ **1**

1.1 Motivasyon 1
1.2 Katkılar 5
1.2.1 Tezde ele alınan yaklaşım 5
1.2.2 Konya Kapalı Havzası durum çalışması 7
1.3 Tezin ana hatları 8

2 KAYNAK ARAŞTIRMASI **11**

2.1 Kartografya, Web Kartografya, Web 2.0 Teknolojisi ve Mashuplar 11
2.2 Web ve Coğrafi Bilgi Sistemleri İlişkisi 14
2.3 Mekânsal Görsel Analitik 16
2.4 Önceki Çalışmalar 18

3 ÇEVRİMİÇİ HARİTALAR: İLGİLİ TEKNOLOJİLER, MASHUP ve API KAVRAMLARI **23**

3.1 İlgili Teknolojiler 25
3.2 Mashup Kavramı 28
3.2.1 Mekânsal veriler için bazı yaygın veri formatları 30
3.2.1.1 GML 30
3.2.1.2 KML 31
3.2.1.3 GeoRSS – GeoRSS GML 32
3.2.1.4 GeoJSON - TopoJSON 33
3.3 API Kavramı 34
3.3.1 Haritalama amaçlı API'ler 36
3.3.1.1 Google Maps API 38
3.3.1.2 BingMaps Control 39
3.3.1.3 MapQuest OpenAPI 39
3.3.1.4 ArcGIS JavaScript API 40
3.3.1.5 OpenLayers JavaScript API 41
3.3.1.6 Carto DB 41
3.3.2 Diğer API'ler 42

4 DURUM ÇALIŞMASI: KKH MASHUP UYGULAMASI **45**

4.1 Çalışma Bölgesi 45
4.2 Kullanılan Veri Seti 46
4.3 Harita Mashup Uygulaması 49
4.3.1 Altlık Harita: Google Maps 49
4.3.2 Grafik ara yüz tasarımı 57

4.3.3 Veri katmanları 63
4.3.4 Dinamik lejant yaklaşımı 66
4.3.5 Zaman kaydırma aracı 68
4.3.6 İstatistiksel yüzey (izoplet harita) yaklaşımı 69
4.3.6.1 Point-in-Poligon testi 73
4.3.6.2 İzoplet harita 79
4.3.7 Katman karşılaştırma (Swipe Mode) ara yüzü 82

5 ARAŞTIRMA BULGULARI VE TARTIŞMA 85

5.1 Mevcut Harita Servislerinin Değerlendirilmesi 86
5.2 Sınırlamalar 89
5.3 Veri Gizliliği ve Kullanım Politikaları 92
5.4 Mashup Uygulamasının Değerlendirilmesi 93

6 SONUÇLAR VE ÖNERİLER 97

KAYNAKLAR 99

ÖZGEÇMİŞ 107

SİMGELER VE KISALTMALAR

Kısaltmalar

AJAX	: Asynchronous JavaScript and XML
API	: Application Programming Interface
CBS	: Coğrafi Bilgi Sistemi
CSS	: Cascading Style Sheet
CSV	: Comma Separated Values
DOM	: Document Object Model
DSİ	: Devlet Su İşleri
ESA	: European Space Agency
ESRI	: Environmental Systems Research Institute
GML	: Geography Markup Language
GNSS	: Global Navigation Satellite System
GPS	: Global Positioning System
HTML	: Hyper Text Markup Language
ICA	: International Cartographic Association
InSAR	: Interferometric Synthetic Aperture Radar
JSON	: JavaScript Object Notation
KKH	: Konya Kapalı Havzası
KML	: Keyhole Markup Language
MGM	: Meteoroloji Genel Müdürlüğü
NASA	: National Aeronautics and Space Administration
NDVI	: Normalized Difference Vegetation Index
OGC	: Open Geospatial Consortium
OSM	: Open Street Map
REST	: Representational State Transfer
RIA	: Rich Internet Applications
ROA	: Resource-Oriented Architecture
RSS	: Really Simple Syndication
SDK	: Software Development Kit
SOA	: Service-Oriented Architecture
SOAP	: Simple Object Access Protocol
SVG	: Scalable Vector Graphics
USGS	: United States Geological Survey
UTM	: Universal Transverse Mercator
W3C	: World Wide Web Consortium
WCS	: Web Coverage Service
WebGL	: Web Graphics Library
WFS	: Web Feature Service
WGS84	: World Geodetic System 1984
WMS	: Web Map Service
WSDL	: Web Service Definition Language
WWW	: World Wide Web

1 GİRİŞ

1.1 Motivasyon

Yeryüzü hareketlerinin izlenmesi çalışmalarında jeoloji (Castelletto ve ark., 2013; Michael ve ark., 2013), jeofizik (Teatini ve ark., 2014), hidrojeoloji (Teatini ve ark., 2011), jeodezi (Raspini ve ark., 2012; Nelson ve ark., 2013; Zhu ve ark., 2013; Ge ve ark., 2014; Ishitsuka ve ark., 2014; Boni ve ark., 2015; Ng ve ark., 2015; Du ve ark., 2016) ve jeoteknik (Liu ve ark., 2010; Nelson ve ark., 2013) alanında çalışan uzmanlar, farklı yöntemler kullanmaktadır. Değişimlerin anlaşılabilmesi ve modellenebilmesi için farklı türde ve farklı kaynaklardan toplanan verilerin (yer kabuğu hareketleri, sismik veri, yeraltı su seviyesindeki değişimler, inklinometrik ve meteorolojik veri gibi) bütüncül bir yaklaşımla analiz edilmesi gerekmektedir. Bu verilerin bilgiye dönüştürülmesindeki anahtar yöntem, analiz edilmeleri, görselleştirilmeleri ve yorumlanmalarıdır. Ancak, büyük ve karmaşık veri setlerinin etkili görselleştirme işlemleri, günümüzde kullanılan basit görselleştirme araçlarıyla pek de mümkün olmamaktadır. Bu amaca ulaşmada mekânsal görselleştirme ve keşfedici veri analizleri önemli bir rol oynamaktadır (Andrienko ve ark., 2003; Andrienko ve Andrienko, 2012). Yeni bir yaklaşım olarak ele alınabilecek görsel analitik, bilgi yönetimi, istatistiksel analizler, karar bilimleri ve daha birçok alanı bütüncül bir yaklaşımla birleştirmektedir (Cook ve Thomas, 2005). Veri setleri içerisindeki bağlantıların oluşturulması ve bilgi çıkarımının direkt olarak sağlanabilmesi ile kapsamlı bakış açıları kazanmada, insan sezgileri ve makine yeteneklerini birleştiren, güncel ve kanıtlanmış bir yaklaşımdır (Keim ve ark., 2008). Mekânsal görsel analitik kavramı ise, görsel analitik kavramından türetilmiş bir kavramdır. Mekânsal görsel analitik, kartografya, coğrafi bilgi bilimi ve geniş çaplı olarak görsel analitik alanındaki yaklaşımları mekana bağlı olarak bir araya getirmektedir (Nelson ve ark., 2015). MacEachren (2013) mekânsal görsel analitiği: "…mekânsal bilgi ile/hakkında anlamlı muhakeme yapabilme yetisini destekleyici analitik/hesaplamalı yöntemler için görsel ara yüzlere odaklanmaktadır" şeklinde yorumlamaktadır. Mekânsal görsel analitik, görsel analitiğin bir alt dalıdır ve büyük boyutlu mekânsal veriden mekânsal bilgi üretiminin yaygınlaşması ve kullanımı konularıyla ilgilenmektedir. Ortaya çıkmasında, karar verme süreçlerini desteklemek için, dinamik ve etkileşimli haritaları kullanan mekânsal görselleştirme araştırmaları

önemli bir rol oynamıştır (MacEachren ve ark., 2004; Andrienko ve ark., 2007). Son yıllarda mekânsal görsel analitik coğrafi bilgi bilimi, coğrafya, kartografya ve ilgili benzer alanlarda dikkat çeken önemli bir araştırma alanı haline gelmiştir. Geleneksel görsel analitik tekniklerinden, insanların bilişsel yetilerini artırmak için büyük ölçüde kartografik görselleştirmeye dayanması açısından, farklılık göstermektedir. Bu nedenle tekrarlı hipotez oluşturma ve doğrulama işlemlerini kolaylaştırmak açısından bütünleşik ve etkileşimli, harita merkezli analitik ortamlara ihtiyaç duyulmaktadır. Bu hedef doğrultusunda, yenilikçi kartografik görselleştirme teknikleri ve yüksek performanslı, bilgisayar destekli, mekânsal analiz araçlarının birleştirilmesi büyük olanaklar sunmaktadır (Zhang ve ark., 2015).

Teknolojide yaşanan gelişmeler, veri ve veri kaynaklarına ulaşma yollarını oldukça kolaylaştırmıştır. Ulaşılan verinin boyutu arttıkça, analiz edilmesi ve içerisindeki gizli bilgi dokularının ortaya çıkarılmasındaki güçlük de artmaktadır (Peuquet ve ark., 2015). Yeryüzü hareketlerinin izlenmesi araştırmalarında olduğu gibi çoklu kaynaklardan alınan veri, çoklu değişkenler içerebilmekte ve hipotez oluşturma, keşfetme ve karar verme süreçlerini zorlaştırmaktadır. Böyle bir veri seti hakkında muhakeme yeteneklerini kolaylaştırmak amacıyla, kullanıcıya kendi verisi ile görsel bir ortamda etkileşim kurabileceği, gelişmiş araçların tasarlanması oldukça önemlidir (Kveladze ve ark., 2015). Coğrafi bilgi sistemleri (CBS) mekânsal verinin etkili bir şekilde yönetilebilmesi, depolanması, analiz edilmesi ve haritalanmasını mümkün kılmaktadır. Ayrıca bu işlemleri sadece mekânsal verinin geometrisi ile ya da konumu ile sınırlı kalmayıp, öznitelik bilgileri için de gerçekleştirebilmekte, bu bilgileri mekânsal veri ile ilişkilendirebilmektedir. Günümüzde ileri seviye CBS çalışmaları, en gelişmiş bilgi ve iletişim tekniklerini kullanan gelişmiş görselleştirme sistemleri ve yüksek bilgisayar yeteneklerinden fazlasıyla etkilenmiştir (Karnatak ve ark., 2012). WWW (World Wide Web) ya da bu kaynakta bundan sonra kısa olarak anılacağı gibi 'Web' ve CBS teknolojilerinin birleştirilmesi, mekânsal verinin kullanımı ve kapsamını köklü bir şekilde değiştirmiştir. Mekânsal veriye etkileşimli olarak erişim, gerçek zamanlı veri entegrasyonu, coğrafi bilgi fonksiyonlarının iyileştirilmesi ve platformdan bağımsız CBS analiz araçlarına erişim gibi önemli olanaklar bu teknolojilerle sunulabilmektedir. Bu birleşimde rol oynayan en önemli gelişme Web 2.0 teknolojisinin 2004 yılında ortaya çıkmasıdır. O'Reilly Media Web 2.0 konferansında (O'Reilly, 2005) tanıtılan bu yeni teknoloji, kullanıcısıyla etkileşim olanağı bulunmayan ve kullanıcılarına, sadece web sitelerini pasif olarak görüntüleyebilme imkanı sunan

sitelerin aksine, diğer kullanıcılarla etkileşim kurma ve web sitelerinin içeriğini değiştirebilme imkanı sunan gelişmiş bir teknolojidir. Bu teknolojiyi kullanan web sitelerine örnek olarak; web-tabanlı topluluklar, web uygulamaları, sosyal ağ siteleri, video paylaşım siteleri, wikiler[1], bloglar ve mashuplar verilebilmektedir (Batty ve ark., 2010). Birçok yenilikçi kaynağın ortaya çıkmasına öncülük eden ve web içeriklerinin birleştirilmesi ve etkileşim tekniklerinin yollarını değişmesine sebep olan, farklı web kaynaklarını bir araya getirebilme özelliğine sahip mashup uygulamaları, web 2.0 teknolojisinin ortaya çıkmasından sonra popüler hale gelmeye başlamıştır. Web geliştirme alanında bir mashup, birden fazla web kaynağından alınan içerikleri kullanarak, yeni bir ara yüzde, tek bir servis altında, görüntüleyebilen bir web uygulaması olarak tanımlanabilmektedir (Flanagin ve Metzger, 2008). Teknolojileri, çevrimiçi fonksiyon kütüphaneleri olarak da ifade edilebilen uygulama geliştirme ara yüzlerine (Application Programming Interface, API) dayanmaktadır ve bu teknoloji web geliştiricileri için genellikle ücretsiz olarak sunulmaktadır. Mashup uygulamaları sıklıkla web haritalama[2] (Kono ve ark., 2009; Peterson, 2012b; Werts ve ark., 2012a; Huang ve Liang, 2013; May ve ark., 2014) ve web kartografya (Field ve O'Brien, 2010; Liu ve Palen, 2010; Poorazizi ve Alesheikh, 2011; Singh ve ark., 2012; Werts ve ark., 2012a; Parker ve ark., 2014) alanında kullanılmaktadır. Web servislerinden en az biri web haritalama servisinden oluşan uygulamalara harita mashup uygulamaları denmekte (Pietroniro ve Fichter, 2006) ve harita mashup uygulamaları söz konusu olduğunda Google, Yahoo, Bing Maps ve benzeri birçok API sağlayıcısına ulaşılabilmektedir. Sunulan bu API hizmetlerinin fonksiyonellik açısından yerine getirdikleri işlemler benzer olsa da en yaygın kullanılanı Google tarafından sunulan API hizmetleridir[3].

Web üzerinde harita alanında yaşanan bu gelişmeler web tabanlı CBS kavramını gün geçtikçe popüler hale getirmiştir. Web 2.0 aynı zamanda, Rich Internet Applications (RIA) aracılığı ile daha iyi insan bilgisayar etkileşimi için gelişmiş mashuplar gibi yeni imkanlar sunmaya başlamıştır. RIA bir web tarayıcısı aracılığıyla sunulan bir plug-in[4] (Adobe Flash/Flex, Java ya da Microsoft Silverlight gibi) olup masaüstü uygulamalara benzer özellikler sunmaktadır (Werts ve ark., 2012b). Bu

[1] Wiki web geliştirme literatüründe, içeriğinin ve yapısının, direkt olarak web tarayıcısı kullanılarak, kullanıcılar tarafından ortak şekilde değiştirilebildiği bir web sitesini ifade etmektedir.
[2] Uluslararası literatürde "Web Mapping" olarak yer alan kavram bu çalışmada web haritalama ya da web-tabanlı harita üretimi olarak isimlendirilmektedir.
[3] Bu istatistik: http://www.programmableweb.com/news/top-10-mapping-apis-google-maps-microsoft-bing-maps-and-mapquest/analysis/2015/02/23 adresinden alınmıştır
[4] Bilgisayar dilinde plug-in, mevcut bir bilgisayar yazılımına spesifik özellikler ekleyen bir yazılım bileşenidir.

uygulamaların kullanılabilmeleri için geleneksel uygulamalarda olduğu gibi herhangi bir yazılım yüklenmesine gerek yoktur. Bir web tarayıcısı (Internet Explorer, Google Chrome, Firefox vb.) kullanarak internete erişebilen tüm kullanıcılar, plug-in eklentisini ekleyerek uygulamaları çalıştırabilmektedir. Vijayalakshmi ve Manonmani (2011) mashup uygulamalarının genel olarak özelleştirilmiş sorgular ya da harita tabanlı görselleştirmeler gibi basit görevler için kullanılmakta olduğunu, bununla birlikte mashup teknolojisinin bilimsel anlamda daha temel, karmaşık ve gelişmiş uygulamalar için kullanılma potansiyeline sahip olduğunu belirtmiştir. Bunu kanıtlar nitelikte, geçen zaman içerisinde zengin içeriklere sahip gelişmiş mashup uygulamaları web ortamında daha çok boy göstermeye başlamıştır örn., bkz.,(Karnatak ve ark., 2012; Peterson, 2012b; Thom ve ark., 2012; May ve ark., 2014; Parker ve ark., 2014). Yazılım bileşenlerinin bir web hizmeti olarak sunulabilmesi ve veri tabanlarının web-tabanlı veriler olarak yayınlanabilmesi, mashup uygulamalarının zenginleştirilmesinde ve web-tabanlı CBS sistemleri olarak uygulanabilmesinde önemli bir rol oynamaktadır. Dangermond (2002), henüz web 2.0 kavramı ve harita mashup uygulamaları ortaya çıkmamışken CBS sistemlerinin, istemci-sunucu mimarisi ile uygulanan veri ve araçlara odaklanmaktan, bir web servisleri modeline evrimleşmekte olduğunu ifade etmiştir. Bu yeni mimaride web, sadece veri iletimi için değil aynı zamanda web servisleri olarak adlandırılan, birlikte çalışabilen, yazılım bileşenleri içerisine yerleştirilebilen, mekânsal analiz işlevselliği için de kullanılmaktadır. Bu bileşenler bir araya getirilerek daha büyük, daha kapsamlı servisler ve/veya uygulamalar oluşturulabilmektedir. Farklı türden ortamlar, sistemler ve veriler arasında birlikte çalışabilirlik, bu web servisleri modelinin uygulanabilmesi için temel ilkedir. Her ne kadar CBS işlevsellikleri kazandırılabilen harita mashup uygulamaları oluşturulabilse de gelişmiş CBS çözümleri sunan güçlü masaüstü uygulamalar ile aralarında fonksiyonellik bakımından daima bir boşluk bulunmaktadır. Bu boşluğun doldurulmasında şüphesiz ki günümüze kadar bu tez de dahil olmak üzere, gerçekleştirilen ve gelecekte gerçekleştirilecek olan çalışmaların rolü büyüktür.

Bu tez çalışmasının temel amacı, mashup mimarisine dayanan ve CBS fonksiyonelliklerine sahip bir mekânsal görsel analitik ortamının değerlendirilmesi, tasarlanması ve geliştirilmesidir. Bu amaçla iki farklı ara yüze sahip bir harita mashup uygulaması oluşturulmuştur. Tüm görselleştirme bileşenleri herhangi bir web tarayıcısında çalışabilmekte ve kullanıcı dostu bir ara yüz sağlamaktadır. Uygulamanın geliştirilmesinde, önceki çalışmalar detaylıca ele alınmış, uygun yöntem ve teknikler

detaylı bir değerlendirmeye tabi tutularak kullanılabilirlik, erişim kolaylığı, kısıtlamalar ve gizlilik politikası yönünden detaylı olarak incelenmiştir.

1.2 Katkılar

Bu bölümde, web geliştirme araçları ve web servisleri kullanılarak harita mashup uygulamalarıyla zamana ve mekâna bağlı mekânsal görsel analitik ortamı oluşturulmasında, bu tez çalışmasıyla sağlanan ana katkılar verilmektedir. İlk olarak harita mashup uygulama mimarisi, ardından bu mimarinin Konya Kapalı Havzası'nda (KKH) zemin çökmelerinin ve nedenlerinin araştırılması durum çalışmasına uygulanması ele alınmıştır.

1.2.1 Tezde ele alınan yaklaşım

Tezin ilk katkısı, mashup uygulaması oluşturulurken ele alınan yaklaşımdır. Bunun için öncelikle web haritalama servisleri ve veri katmanları konusuna odaklanılmıştır. Web üzerinde gelişmiş harita mashup uygulamaları oluşturulmasına imkân sunan ve ileri düzey fonksiyonelliklere sahip web haritalama servisleri sistematik bir yaklaşımla ele alınmış, ara yüz geliştirme yaklaşımı beş ana katmana dayanarak tasarlanmıştır. Bu katmanlar, programlama kolaylığı açısından örnek dokümantasyonlara ulaşım, ücretsiz kullanılabilirlik açısından sınırlamalar, dış veri entegrasyonu açısından veri katmanları eklenebilme yetenekleri ve desteklenen veri formatları, paylaşım açısından gizlilik politikası ve kullanım kolaylığı açısından sunulan servisin olgunluğu (yayında olma süresi) şeklindedir.

Çalışmaya web haritalama servislerinin yaygın kullanımları ve kullanıcı topluluklarının araştırılması ile başlanmıştır. Bu amaçla web üzerinde günümüze kadar geliştirilen mashup uygulamaları ve bu uygulamaların harita tabanlı olarak üretilenlerinin sayısına ulaşılmaya çalışılmıştır. Yeni bir harita mashup uygulaması oluşturulurken yararlanılacak örnek uygulamalar ve dokümantasyon sayısıyla bu bilginin orantılı olabileceği düşünülmüştür. Elbette ki web üzerinde şimdiye kadar oluşturulan harita mashup uygulamalarının gerçek sayısına ulaşmak, tamamının kayıt altına alınmaması açısından neredeyse imkânsızdır. Ancak oluşturulan mahsup uygulamalarını kayıt altına alan ve bu uygulamalara ulaşılabilecek bağlantılar ile birlikte istatistiksel bilgilerini de sunan Programmableweb (2016) benzeri özel web

sitelerine ulaşmak mümkündür. Bu ve benzeri kaynaklardan toplanan istatistiksel bilgilerin yanında, harita mashup uygulamaları alanında gerçekleştirilen önemli çalışmalar (Newman ve ark., 2010; Nuojua, 2010; Poorazizi ve Alesheikh, 2011; Sukic ve Rancic, 2011; Bearman ve Appleton, 2012; Bedrina ve ark., 2012; Daniel ve ark., 2012; Karnatak ve ark., 2012; Maso ve ark., 2012; Patel ve ark., 2012; Peterson, 2012b; Singh ve ark., 2012; Werts ve ark., 2012a; Zhang ve ark., 2012; Al-Ahmadi ve ark., 2013; Han ve ark., 2013; Huang ve Liang, 2013; May ve ark., 2014; Parker ve ark., 2014) da incelenmiştir. Kullanılan harita servisleri, sistem mimarisi ve sağlanan öneriler dikkate alınarak incelenen çalışmalardan elde edilen sonuçlar paylaşılmıştır.

Web üzerinde sunulan neredeyse tüm haritalama servisleri, kitle kaynaklı[5], yani verisini kullanıcılarının sağladığı Open Street Map[6] (OSM) gibi özel istisnalar hariç, ticari amaçlı oluşturulan servislerdir. Kitle kaynaklı servislerin veri doğruluğu ve kalitesi açısından, doğruluk gerektiren işlerde kullanılması Misra ve ark. (2014) da belirttiği gibi halen tartışmaya açıktır. Bu nedenle bu gibi servisler, zaman ve konum olgularının ön planda olduğu bu çalışmada, araştırma konusu dışında bırakılmıştır. Diğer web kaynaklı servisler, genelde ticari bir amaca hizmet etmeyen kullanıcılar için, bazı kısıtlamalar haricinde tamamen ücretsiz olarak kullanılabilmektedirler. Ayrıca bu haritalama servislerinin Google Fusion Tables[7] gibi tablosal ya da mekânsal (konum bilgisi içeren) verilerin depolaması, yönetimi ve paylaşımı için geliştirdiği kullanışlı platformlara da ulaşmak mümkündür. Tüm bu servislerdeki kısıtlamalar, oluşturulacak sistem mimarisini doğrudan etkileyebilmeleri açısından açık bir şekilde tanımlanmalıdır. Oluşturulan mashup uygulaması için kullanılan servislerdeki kısıtlamalar da detaylıca ele alınmıştır (bkz. bölüm 5).

Web servisleri kullanılarak web-tabanlı CBS benzeri harita mashup uygulamaları oluşturulmasındaki en önemli konuların başında, servisin, herhangi bir dış kaynaktan alınan verinin entegrasyonunda sunduğu yeteneklerdir. Buradaki temel mantığın, kullanıcıların kendi verisini esas alarak web-tabanlı harita servisleri ile kendi özel harita uygulamaları amaçlandığında, servisin sağladığı desteğin önemi daha iyi anlaşılabilmektedir. Bir diğer önemli konu da desteklenen veri formatlarıdır. Bu alanda sağlanan esneklik, uygulamalarda kullanılacak verinin, istenen formatlara dönüşümü için harcanacak çaba ve zaman açısından önemli avantajlar sağlayacaktır. Kullanılan

[5] Kitle kaynaklı veri (orj. crowd-sourcing) internet kullanıcıları tarafından kişisel cihazlarla (akıllı telefonlar ve diğer akıllı cihazlar gibi) üretilen veriyi temsil etmektedir.
[6] Detaylı inceleme için bkz. https://www.openstreetmap.org
[7] Google Fusion Tables kullanım ve detaylar için bkz. https://www.google.com/fusiontables

servisler, veri katmanları ve desteklenen veri türleri açısından değerlendirilmiş, gerçekleştirilen durum çalışması ile bu konuda önemli yönlendirmeler önerilmiştir.

Kişisel verilerle oluşturulan harita mashup uygulamalarında, kişisel veri ve bilgilerin gizliliği ön plana çıkmaktadır. Ayrıca servislerin kullanım ve gizlilik politikaları, bu servisler ile yeni uygulamalar oluşturma aşamasında, kullanıcıyla servis sağlayıcısı arasında kabul edildiği varsayılan bir sözleşme yerine geçmektedir. Her servis sağlayıcısının kendine özgü kullanım şartları ve gizlilik politikaları mevcuttur. Ayrıca sunulan hizmetler, haritalar ve veriler bir takım telif haklarına tabi tutulabilmektedir. Bu nedenle servislerin kullanım koşullarının, gerçekleştirilecek uygulamanın gereksinimlerini karşılayıp karşılamadığı önceden belirlenmelidir. Çoğunluğu ticari amaçlı hizmet sunan bu servisler, kişisel bilgilerin hangi koşullarda başka amaçlar için kullanılabileceği ve üçüncü kişilerle paylaşabileceği konusunda detaylı bilgi sunmakla yükümlüdür.

Son olarak sunulan hizmetin yayınlanmaya başlandığı tarih, kullanım yaygınlığı, gerçekleştirilen hata ayıklamaları, kullanıcı geribildirimleri ile gerçekleştirilen güncellemeler (stabilite) ve kullanıcıların bu servisleri kullanmaya aşinalığı, servisin olgunluğu olarak isimlendirilmiştir. Bir servisin olgunluk derecesi hem sürdürülebilir olduğu hem de gelecekte geliştirilecek yeniliklere ayak uydurması açısından oldukça önemlidir. Sonuçta her bir servis sağlayıcısının bir takım ticari amaçlar doğrultusunda sunduğu hizmetleri planlaması ve bu amaçlarla hizmetini sürdürmesi gerekmektedir. Servisin bir süre sonra sona erdirilmesi ya da köklü değişikliklere uğratılması, oluşturulacak uygulamada aksaklıklara neden olacaktır.

Bu beş katman, harita mashup uygulaması yaklaşımının temelini oluşturmaktadır. Web ortamına taşınan tüm harita mashup uygulamaları bu katmanların hepsine ihtiyaç duymayabilir ancak web teknolojileri sayesinde, geliştirilecek CBS benzeri mashup uygulamaları için bir model olabileceği düşünülmektedir.

1.2.2 Konya Kapalı Havzası durum çalışması

Tezin diğer katkısı da tezde ele alınan yaklaşımın harita mashup uygulaması geliştirilerek, KKH'da zemin çökmeleri ve nedenlerinin araştırılması durum çalışmasına uygulanmasıdır. Bunun için Google Map API V3[8], Google Fusion Tables REST

[8] Google Maps API V3 için https://developers.google.com/maps/

(REpresentational State Transfer) API[9], Highcharts API[10], AJAX[11] (asynchronous JavaScript and XML) web geliştirme teknikleri ile Google Drive[12] servislerini kullanmaktadır. Uygulama aynı zamanda esneklik ve daha fazla etkileşim için HTML (HyperText Markup Language), CSS (Cascading Style Sheet) ve JavaScript gibi web istemci teknolojilerini de kullanmaktadır. Bu teknolojiler kullanılarak web-tabanlı harita servislerinin hali hazırda sunmadığı, dış verinin katman olarak eklenmesinde katman gruplandırma ve eklenen katmanların dinamik lejant alanlarının oluşturulması problemine bir çözüm geliştirilmiştir. Ayrıca genel amaçları konum paylaşmak, konum tabanlı aramalar ve reklam yayınları olan web harita servisleri ile gelişmiş mashup uygulamaları oluşturulması konusunda, izoplet veri katmanlarının oluşturulması problemine yeni bir yaklaşımla çözüm sunulmaya çalışılmıştır. "Point in Polygon" ve "Multikuadrik Radyal Bazlı Fonksiyon" yaklaşımının kullanıldığı çözüm ile kullanıcılar, istatistiksel yüzeyleri oluşturabilmekte ve web-tabanlı harita servislerine katman olarak ekleyerek izoplet haritalar oluşturabilmektedir. Harita üzerine eklenen katmanların zamana bağlı olarak değişimlerinin izlenmesi için "Swipe Mode" adı verilen yeni bir ara yüz tasarlanmıştır. Bu ara yüz ile iki farklı katman harita üzerine eş zamanlı olarak eklenebilmekte ve geliştirilen ekran süpürme aracıyla katmanlar arasındaki değişimlerin izlenmesi sağlanmaktadır. Zamana bağlı değişimlerin izlenmesi için uygulamaya eklenen bir diğer yaklaşım ise zaman kaydırma çubuğu aracıdır. Bu araç, harita üzerine eklenecek katmanların zamana bağlı olarak çağırılmasını sağlamaktadır. Oluşturulan tüm görselleştirme bileşenleri herhangi bir web tarayıcısında çalışabilmekte ve kullanıcı dostu bir ara yüz sağlamaktadır. Tüm bu katkıların bir araya getirilmesi ve hayata geçirilmesi ile web-tabanlı harita mashup uygulamaları ve güçlü masaüstü CBS uygulamaları arasındaki köprünün inşası için bir yapı taşı daha ekleneceği düşünülmektedir.

1.3 Tezin ana hatları

Tez, yukarıda açıklanan katkıları gerçekleştirmek üzere yapısal olarak aşağıdaki gibi kurgulanmıştır: 2. bölümde kaynak araştırmasına yer verilmiş, tez çalışmasının altyapısını oluşturan yöntem, teknik ve araştırmalarla ilgili önceki çalışmalar ele

[9] Google Fusion Tables REST API için https://developers.google.com/fusiontables/
[10] Highcharts API için http://www.highcharts.com/
[11] AJAX ile ilgili detaylı bilgi için https://tr.wikipedia.org/wiki/AJAX_(programlama)
[12] Google Drive için https://www.google.com/drive/

alınmıştır. Bu amaçla bu bölümde detaylandırılan literatür özeti, zamana ve mekana bağlı çok parametreli olguların, mevcut web haritalama servisleri aracılığıyla etkili ve etkin görsel sunumlarının, web tabanlı CBS benzeri bir sistem oluşturularak sunulması bakış açısıyla; web teknolojisinin ortaya çıkışı, web üzerinden yayınlanan haritaların harita üretimine etkisi, Web 2.0 teknolojisinin duyurulması ve web tabanlı haritalarda yaşanan gelişmeler, web ve CBS sistemlerinin ilişkisi, zamana ve mekana bağlı olguların modellenmesinde görsel analitik yaklaşımı, mekânsal görsel analitik kavramı ve web tabanlı CBS sistemlerinde bu yaklaşımın kullanılabilirliği ile tüm bu yaklaşım ve teknolojilerin bir araya getirilerek kullanıldığı önceki çalışmaların incelenmesi, sıralı kurgusuna dayandırılmıştır.

Üçüncü bölümde web haritalama servisleri kullanılarak oluşturulacak gelişmiş uygulamalar için gereken anahtar teknolojilerin tanıtılması ve mevcut web harita servislerinin değerlendirilmesi amaçlanmıştır. İlk olarak web geliştirme araçları, mashup kavramı ve web haritalama faaliyetlerinde kullanılan veri değişim formatları ele alınmıştır. Harita mashup uygulamalarına genel bir bakış yapılarak harita mashup uygulamalarındaki potansiyel ortaya koyulmaya çalışılmıştır. Mashup kavramı, kavramın tarihçesi, kullanım alanları ve haritalama amaçlı mashup uygulamalarının diğer mashup uygulamaları arasındaki yeri belirlenmeye çalışılmıştır. Ardından mashup uygulamalarının oluşturulmasında kullanılan API ara yüzleri ve haritalama amaçlı olmayan ancak harita mashup uygulamalarında sıklıkla kullanılan diğer API ara yüzleri detaylandırılmaya çalışılmıştır. Haritalama alanında kullanılan API ara yüzleri yaygın kullanım, potansiyel, dokümantasyon, kullanım kolaylığı ve işlevsellik kıstaslarına göre değerlendirilmiştir. Bölümün temel amacı, bu tezde kullanılacak uygun teknolojilerin ve araçların belirlenmesinde yönlendirici olmasıdır.

Dördüncü bölümde, KKH mashup uygulaması durum çalışması ele alınmıştır. İlk olarak çalışma bölgesi tanıtılmıştır. Ardından bu bölgede zemin çökmeleri ve nedenlerinin araştırılması için kullanılacak veri seti detaylandırılmıştır. Oluşturulan harita mashup uygulaması; 1) grafik ara yüz tasarımları, 2) veri katmanları, 3) dinamik lejant yaklaşımı, 4) zaman kaydırma çubuğu yaklaşımı, 5) istatistiksel yüzey (izoplet harita katmanı) yaklaşımı ve 6) katman karşılaştırma “swipe mode” ara yüzü, başlıkları altında yine bu bölümde ele alınmıştır.

Beşinci bölümde araştırma bulguları ve tartışma konularına yer verilmiştir. Gerçekleştirilen çalışma sürecinde karşılaşılan zorluklar, kullanılan servislerin

kullanılabilirlik, erişim kolaylığı, kısıtlamalar ve gizlilik politikası yönünden değerlendirilmeleri bu bölümde tartışılmıştır.

Altıncı bölümde çalışma ile gerçekleştirilen katkılar genel olarak özetlenmiştir. Ayrıca harita mashup uygulamaları konusunda gelecekte gerçekleştirilmesinin mümkün olduğu düşünülen bazı önemli fikirler paylaşılmıştır.

2 KAYNAK ARAŞTIRMASI

Bu bölümde detaylı literatür özeti incelemesi yapılmıştır. Tezin amacı da dikkate alınarak literatür taraması, zamana ve mekana bağlı olguların, mevcut web haritalama servisleri aracılığıyla, etkili ve etkin görsel sunumlarının, web-tabanlı CBS sistemi oluşturularak sunulması konularını kapsayacak şekilde oluşturulmuştur. İlk olarak temel güncel kavramlar, ardından web teknolojisinin ortaya çıkışı ve web üzerinden yayınlanan haritaların harita üretimine etkisi incelenmiştir. Web-tabanlı harita üretimi ve web-tabanlı CBS teknolojilerinin temel yapısını oluşturan Web 2.0 teknolojisi yine bu bölümde detaylıca ele alınmıştır. Bu amaçla, Web 2.0 teknolojisinin duyurulması ile web tabanlı haritalarda yaşanan gelişmeler ve web tabanlı CBS sistemlerinin gelişimi konuları, incelenen kaynaklardan derlenerek detaylandırılmıştır. Son olarak, zamana ve mekâna bağlı olguların modellenmesinde mekânsal görsel analitik yaklaşımı, web tabanlı CBS sistemlerinde bu yaklaşımın kullanılabilirliği ve tüm bu yaklaşım ve teknolojilerin bir araya getirilerek kullanıldığı önceki çalışmaların incelenmesi sıralı kurgusu ile kaynak araştırması sonuçlandırılmıştır.

2.1 Kartografya, Web Kartografya, Web 2.0 Teknolojisi ve Mashuplar

Uluslararası Kartografya Birliği 2011-2019 stratejik planında (International Cartographic Association, 2011) herhangi bir değişiklik yapmadığı ve 2003-2011 stratejik planında (International Cartographic Association, 2003) *"harita tasarım, üretim ve kullanım sanatı, bilimi ve teknolojisi ile ilgilenen bilimdir"* şeklinde kısa tanımını verdiği kartografya, aynı kaynakta *"coğrafi uzayın görsel ya da sanal temsillerinin – haritalar – üretimi ve manipülasyonu ve bu uzay hakkındaki bilgilerin keşfi, analizi, anlaşılması ve iletişimi için benzersiz bir vasıtadır"* şeklinde uzun tanımı ile yer almaktadır. Bu tanımlardan anlaşılacağı gibi, coğrafi uzayın birer tasviri olan haritalar kartografyanın merkezinde yer almaktadır. Günümüzde, üretildikleri ortam dikkate alınarak, basılı (hardcopy), bir bilgisayar ekranında görüntülenebilen, bir depolama aygıtında depolanan ya da bir veri tabanı vasıtasıyla bir ağ üzerinden erişilebilen, gibi birçok farklı biçimde haritanın varlığından söz edilebilmektedir. Web kartografya tanımında kartografya için yapılan tanımdan herhangi bir farklılık söz konusu değildir. Sadece burada ortam web ile sınırlıdır (Kraak, 2001).

Web teknolojisinin günümüzdeki haline gelebilmesi için büyük bir evrim geçirdiği varsayılırsa, bu evrimin ilk evreleri Web 1.0 olarak adlandırılabilmektedir (Wikipedia, 2016b). Eick (2010) Web 1.0[13] teknolojisinin, ilk nesil internet tarayıcılarının uygulama ve programları ile ilgili bir kavram olduğunu ifade etmektedir. Bu teknoloji istemci tarafı ya da sunucu tarafında oluşturulan girdileri desteklemek ve bilgiye ulaşmak için kullanılmaktadır. Bu gibi bir teknoloji, sistem kullanıcılarına zengin ve etkili bir deneyim oluşturmaktan ziyade, uygulanabilir bir sistem oluşturmak için geliştirilmiştir. Bu nedenle geleneksel web 1.0 teknolojileri makineler için verimli sistemler sayılabilmelerine rağmen kullanıcılar açısından pek verimli sistemler olduğu söylenememektedir. Web 1.0, "tıkla-bekle" ya da "çalıştır-durdur" felsefesi ile çalışmaktadır. Bu, bir kullanıcının, herhangi bir web sayfasını görüntülemek istediğinde, istemci tarafındaki internet tarayıcısını kullanarak bir sorgu üretmesi ve daha sonra bu sorgunun sunucuya gönderilmesi gerektiği anlamına gelmektedir. Kullanıcı daha sonra, sunucunun sorguyu cevaplamasını ve cevabı tekrar istemci tarafına göndermesini beklemek zorundadır. Kullanıcı ayrıca görmek istediği web sayfasını yükleyebilmek için, gerçekleştirdiği tüm sorguların cevabının internet tarayıcısına ulaşmasını ve cevaplanan sorgulamaların web tarayıcısı tarafından gösterilmesini de beklemek zorundadır.

Bu işlemler, çoğu kullanıcı için zaman kaybına yol açmaktadır. İlk olarak yavaş ya da sınırlı erişime sahip İnternet bağlantılarında yapılan bir dizi sorgulamanın, sunucu tarafına iletilmesi ve cevaplanması oldukça uzun bir zaman alabilmektedir. Ayrıca kullanıcı yüksek hızlı internet bağlantısına sahip olsa dahi, harita uygulamaları, spor müsabakalarının canlı sonuçları ya da karşılıklı oyunlar gibi uygulamalara sahip web sayfalarında, verilerin ya da web sayfasının tamamının sürekli güncellenmesi gerekmektedir ve bu durum da önemli gecikmelere sebep olmaktadır. Bunun sebebi, web sayfasındaki görüntülenen bilgilerdeki küçük değişimler için dahi, sayfanın tamamının tekrar yenilenmesini gerektiren Web 1.0 gereksinimlerinden kaynaklanmaktadır (Cormode ve Krishnamurthy, 2008).

İnternete erişmek ve erişimleri geliştirmek amacıyla ihtiyaç duyulan yeni nesil teknolojiler "Web 2.0" başlığı altında toplanmakta ve geliştirilmektedir. Web 2.0 kavramı Tim O'Reilly Media (2005) Web 2.0 konferansı ile tanıtılmıştır. Bu kavram,

[13] Web 1.0 kavramı aslında karşılığı olmayan bir kavramdır. Web 2.0 teknolojisinin tanıtılmasından sonra World Wide Web teknolojisinin ilk evrelerini temsil etmek için kullanılan bir retronimdir. Detaylı bilgi için bkz. https://en.wikipedia.org/wiki/Web_2.0

yeni bir web sürümü gibi isimlendirilse de, herhangi bir teknik özellikte yapılan güncellemeyi işaret etmemekte, daha ziyade yazılım geliştiricilerin ve son kullanıcıların webi kullanma yollarında gerçekleşen toplu değişikliği ifade etmektedir. Web 2.0 kavramının anahtar özelliği, daha önce ifade edilen "tıkla-bekle" ya da "çalıştır-durdur" döngüsünün, ilgili web sunucusundan, kullanıcıya özel bir web sayfasıyla, ilgili verinin eş zamansız olarak sağlanmasıyla elimine edilmesidir (Cormode ve Krishnamurthy, 2008). Bu transfer işlemi, kullanıcı web sayfasını görüntülemeye devam ederken ya da etkileşim içindeyken, bir arka plan görevi olarak gerçekleştirilmektedir. Web 2.0 teknolojisi ile birlikte 'mashup' kavramı da gündeme gelmiştir. Konu web geliştirme ve anonim bir web terimini açıklamaya geldiğinde, her ne kadar akademik camiada haklı olarak[14] pek kabul görmese de, Wikipedia'dan daha iyi tanımların bulunması zordur. Wikipedia (2016c) bir web mashup uygulamasını *"bir ya da daha fazla kaynaktan alınan içerikleri bir araya getiren bir web sitesi ya da web uygulaması"* olarak tanımlamaktadır. Mashup kavramının kökü müzik endüstrisine dayanmaktadır. Müzikte bu terim, farklı parçalardan alınan vokal ya da enstrümental parçaların karıştırılması ile yeni bir parçanın oluşturulduğu bir popüler müzik türünü temsil etmektedir. Mevcut seslerin alınarak farklı bir ses şeklinde bir araya getirilmesi, görsel iletişim için de ilham kaynağı olmuştur (Pietroniro ve Fichter, 2006). Bu dönüşüm işlemi haritalar ve mekânsal içerikler için 'Harita Mashup Uygulamaları' biçiminde gerçekleşmiştir. Genel anlamıyla kullanıcı odaklı bu yeni teknolojiler ve özellikle harita mashup uygulamalarının hızlı gelişiminin kaynağı, Web 2.0 teknolojisinin yükselişi ile yakından ilgilidir.

Bu gelişmelerin ardından Google, Microsoft Search (Bing) ve Yahoo! gibi en popüler web arama motorları, kendi özel coğrafi ara yüzlerini geliştirerek, standart web tarayıcılarında çalışabilen ve ilk zamanlarda sadece uzmanların ve seçilmiş kullanıcı topluluklarının erişebildiği detaylı uydu görüntüleri ve hava fotoğraflarını, genel kullanıcıların kullanımına açarak, konum-tabanlı arama alanına katılmıştır. Bu şirketler kullanıcılarına konuma dayalı aramalar yapabilecekleri çevrimiçi haritalar sunmaya başlamıştır. Çevrimiçi haritalar sunan Internet siteleri her zaman internet kullanıcılarının ilgi odağında olmuştur. Bununla birlikte Internet kullanıcıları bu servislerin hayata geçirilmesinin ardından, haritaları sadece görüntülemekle kalmayıp

[14] Wikipedia tüm internet kullanıcılarının değiştirebildiği içeriğe sahip olmasından ötürü bu sitede yer alan referanslanmamış bilgiler çarpıtılmış, eksik yazılmış ya da telif haklarına tabi olabilmektedir. Bu durum internet sitesinde de uyarı olarak sunulmaktadır.
Ayrıntılı bilgi için https://en.wikipedia.org/wiki/Wikipedia:Citing_Wikipedia adresine başvurulabilir.

artık çok daha fazlasını yapabilmektedirler. Pietroniro ve Fichter (2006) bu servislerin duyurulmasının hemen ardından, harita mashup uygulamalarını, nasıl kullanılmaya başlandığını ve haritalar ile harita üreticileri üzerindeki etkilerini araştırmıştır. Onlara göre geniş çaplı Internet erişim olanakları ve ucuz bilgisayarların kullanımının artmasıyla insanların çevrimiçi harita hizmeti sunan servislere ilgisi artmış ve web-tabanlı haritaların üretim ve kullanım şekilleri hızlı bir şekilde değişmiştir. Haritalar çok daha fazla detaylı üretilmeye başlanmış ve statik objelerden dinamik olayların bir araya getirildiği bilgi sistemlerine dönüşmeye başlamıştır.

Web-tabanlı haritalama servisleri kullanılarak harita üretiminde, Google'ın 2005 yılında, geniş spektrumlu programlama düzeylerine bağlı olarak, farklı veri ve yazılımları farklı uygulamalar için eşleştirme ve bir araya getirme imkânı sunan API (Application Programming Interface) ara yüzünü geliştirmesinin payı büyüktür. Web Services Description Language (WSDL) olarak bilinen ve bir makine tarafından okunabilen formatta yazılan web servis API ara yüzleri, web uygulamalarının birbiri ile iletişim kurması için takip etmesi gereken yönergeleri içermektedir (Han ve ark., 2013). Google Maps API kullanırken, programlamacılar farklı türde hazır işlevselliklere ya da sınıflara erişebilmekte ve kendi dış verilerini kullanarak işlemler gerçekleştirmek için, bu sınıfları kullanarak, kendi özel uygulamalarını oluşturabilmektedir. Google Maps API aslında, bir etkileşimli haritanın çeşitli elemanlarını oluşturmak için bir web sayfası tarafından çağırılan, bir dizi JavaScript sınıfından oluşmaktadır. Bu ara yüzün geliştirilmesinden sonra Internet üzerinde günümüze kadar oluşturulan harita mashup uygulaması sayısını bulabilmek neredeyse[15] imkânsızdır. Ağustos 2008'de harita tabanlı mashup uygulamalarının sayısı 1740 iken (Macdonald, 2008) 2010 Şubat ayında bu sayı 2153 olarak (Batty ve ark., 2010), günümüzde ise 4556 olarak elde edilmiştir (Programmableweb, 2016).

2.2 Web ve Coğrafi Bilgi Sistemleri İlişkisi

Web ve CBS ilişkisinin iyi anlaşılabilmesi için öncelikle temel kavramlara odaklanmak gerekmektedir.

[15] Web geliştiricileri tarafından oluşturulan mashup uygulamalarının sayısını bulmak, bu uygulamaları oluşturan geliştiricilerin, uygulamalarını belirli istatistik verisi toplayan sitelere kayıt ettirmemeleri durumunda imkânsızdır. Bu sitelere en popüler örnek olarak http://www.programmableweb.com/ verilebilir.

Bir harita, Uluslararası Kartografya Birliği (2003) tarafından: *"seçilmiş ayrıntılar ve karakteristikleri temsil eden, yazarının yaratıcı seçimleriyle şekillenen ve mekânsal ilişkiler birincil öneme sahip olduğu durumlarda kullanılmak üzere tasarlanan, coğrafi gerçekliğin bir tasviri"* olarak tanımlanmaktadır. Aynı kaynakta coğrafi bilgi bilimi (bu çalışmada CBS olarak kullanılmıştır): *"mekânsal bilginin işlenmesi ve yönetimi ile bunlarla ilişkili teknolojileri içeren bilimsel araştırma alanıdır. Bilgi işleme ve yönetimi veri analizi ve dönüşümleri, veri yönetimi ve bilgi görselleştirme alanlarını içermektedir"* şeklinde tanımlanmaktadır.

Kraak (2015) 10-12 Kasım 2015 tarihinde Viyana Teknik Üniversitesinde gerçekleştirilen Uluslararası Kartografya Birliği Avrupa Kartografya Sempozyumunun[16] açılış konuşmasında günümüz olanaklarını da dikkate alarak bu temel kavramlar konusunda şu ifadelere yer vermiştir:

"Haritalar, toplumumuzda benzersiz ve önemli bir rol oynamaktadır. Haritalar mekânsal verilerimiz için bir ara yüzdür. Yolumuzu bulmak, veri dokuları ve eğilimleri kavramayı sağlamak ve karar verme işlemlerini desteklemede önemli birer yardımcıdır. Aynı zamanda eğlencelidir, güzel tasarımlanabilmektedir ve uzak, egzotik yerler hakkında hayal kurabilmemizi sağlayabilmektedir. Bu işlemleri mümkün kılan disiplin ise kartografyadır. Kartografya, haritaların tasarımı, üretimi ve kullanımı sanatı, bilimi ve teknolojisidir. Son yıllarda gerçekleşen bilimsel ve teknolojik gelişmeler harita üretimi ve kullanımı seçeneklerini bir adım öteye taşımıştır. İnternet, bir masaüstü ya da mobil cihaz aracılığıyla, haritaları gezegenimizin her bir köşesine dağıtabilir hale gelmiştir. Geniş bir yelpazeye sahip veri ve araçların kullanılabilir olmasından bu yana, herkes harita üretebilme potansiyeline kavuşmuştur. Bununla birlikte iletişimsel haritaların tasarımı halen bir sanattır ve kartografya bilimine ihtiyaç duymaktadır".

Burada vurgulanmak istenen konu, internet olanaklarının, tüm kullanıcılarını birer harita üreticisine dönüştürdüğüdür, ancak konu harita ile iletişime geldiğinde mutlaka kartografya bilimine başvurulması gerektiğidir.

Web-tabanlı harita üretimi kavramı, şimdiye kadar açıklanan tanımlar ve ifadeler göz önüne alınarak, web üzerinde haritalar, mekânsal veri ve web harita servislerinin tasarımlanması, uygulanması, üretimi ve dağıtımı sürecini temsil eden bir kavramdır (Wikipedia, 2016a) şeklinde yorumlanabilmektedir. Li ve ark. (2011) web CBS kavramı ve web-tabanlı harita üretimi kavramlarının çoğu zaman birbirlerinin yerine

[16] ICA European Symposium on Cartography (http://www.eurocarto.org)

kullanılabilen kavramlar olduğunu savunmaktadır. Onlara göre bu iki kavram aynı olguyu ifade etmektedir. Bu yaklaşım, web CBS kavramının, web tabanlı haritalara mekânsal verinin işlenmesini destekleyici mekânsal analiz fonksiyonellikleri eklenmesiyle elde edildiği düşünüldüğünde, gerçekçi bir yaklaşımdır. Web haritaları ve CBS, özellikle son yirmi yılda statik haritaların basit olarak görselleştirilmesinden, daha dinamik ve yarı-dinamik[17] veri görselleştirme ve veri analizine, daha yakın tarihte ise veri ve harita servislerine doğru çarpıcı bir evrim geçirmiştir. Önceki bölümde sözü edilen yeni web teknolojilerinin kullanılmasıyla devlet kurumları, özel şirketler, araştırma enstitüleri ve açık kullanıcı toplulukları tarafından birçok web haritalama ve web CBS teknolojisi geliştirilmiştir (Li ve ark., 2011). Bu teknolojiler; mekânsal veri erişimi ve yayımı; mekânsal veri keşfi ve görselleştirilmesi; mekânsal veri işleme, analizi ve modelleme; halk katılımcı[18] CBS kullanan ortak mekânsal karar destek sistemleri ve web-tabanlı mekânsal servislerin entegrasyonu işlemleri için uygulamalar oluşturmada kullanılmaktadır (Li, 2008). Web ve web haritaları varlığını sürdürdüğü sürece, bu kavramlarla sıkı sıkıya bağlı web CBS kavramı da daima gündemde olacaktır.

2.3 Mekânsal Görsel Analitik

Ele alınan tüm bu konuların yanında, web tabanlı haritalama servisleri kullanılarak oluşturulan bir web CBS içerisinde, zamana bağlı olguların gösterimi, ayrı olarak değerlendirilmesi gereken bir konudur. Öncelikle mekânsal olguların zamana bağlı değişimlerini etkili şekilde ele alan yaklaşımlara ihtiyaç duyulmaktadır. Görselleştirme işlemi uzun zamandan beri zamana bağlı verilerin analiz edilmesine başarılı bir şekilde uygulanmaktadır.

Son zamanlarda özellikle büyük boyutlu ve çok parametreli veri setleri için, daha etkili grafikler sağlayan birçok yeni yaklaşım ortaya atılmıştır (Nguyen, 2015). Zaman parametresine sahip verilerinin analizi bilim, mühendislik ve ticaret alanlarında en fazla ortaya çıkan problemlerden biridir. Bu problem, bilim ve mühendislik alanlarında daha hassas alıcıların ortaya çıkması ve kuruluşlarda bilgisayar kullanımının

[17] Burada yarı-dinamikten kasıt kullanılan altlık harita ya da harita üzerinde gösterilen veriden herhangi birinin statik olması durumudur.

[18] Kökeni "Public participation geographic information systems (PPGIS)" olan kavram mekânsal bilgi üretimini desteklemek amacıyla akademik uygulamaları yerel düzeye indirme maksadıyla ortaya çıkmıştır. Burada halk tarafından üretilen mekânsal veri esastır.

artması gerçeğine bağlı olarak, büyük boyutta toplanan zamana bağlı verilerde yaşanan artışla birlikte, daha çok önem kazanmıştır. Uzun zamandır görselleştirme işlemi zamana bağlı verilerinin analizinde etkili bir yaklaşım olduğunu kanıtlamıştır. Bu yaklaşımın ardındaki motivasyon, görüntüler içerisindeki yapıların fark edilebilmesi için, insan gözünün olağanüstü yeteneklerinden faydalanmaktır. Ancak verinin basit görselleştirme teknikleri ile salt olarak görselleştirilmesi bazı durumlarda veri içerisindeki dokuların tam olarak keşfedilememesine neden olmaktadır. Özellikle çoklu kaynaklardan toplanan büyük boyutlu veri seti söz konusu olduğunda bu problem ortaya çıkmaktadır. Böylesine bir problemin üstesinden gelebilmek için bilim dünyası yakın bir tarihte ortaya çıkan görsel analitik ve mekânsal görsel analitik konularına odaklanmaktadır.

Görsel analitik, iteratif olarak bilgi toplayan, veriyi ön işlemlere tabi tutan, istatistiksel analizler gerçekleştiren (Peck ve ark., 2015), veri madenciliği yapan ve makine öğrenimi (Witten ve Frank, 2005), bilgi gösterimi (Van Harmelen ve ark., 2008), kullanıcı etkileşimi (Yvonne ve ark., 2011), görsel sunumlar (Lee ve Rojas, 2013; Garreau ve ark., 2015; Leopold, 2015), insan muhakeme teknikleri (Drigas ve ark., 2011), algılama, keşfetme ve insan yeteneklerini, karar verme işleminde kullanan (Keim ve ark., 2006; Llorà ve ark., 2006) bir süreçtir. Bu yaklaşım kuş gribi virüsünün zamana bağlı olarak yayılımının incelenmesi (Proulx ve ark., 2006), şehir içi akıllı ulaşım sistemi oluşturulması (Liu ve ark., 2013), sosyal ağların analizi (Ghani ve ark., 2013), zamana bağlı olayların incelenmesi (Steed ve ark., 2013; Wiringa ve ark., 2013; Kveladze ve ark., 2015; Nelson ve ark., 2015) ve hatta sosyal ağlar üzerinden paylaşılan politik söylemlerin takibi (Nelson ve ark., 2015) gibi birçok farklı probleme uygulanmıştır. Bu nedenle, büyük boyutlu ve çok değişkenli veri setlerinden fikir edinmede analiz yapmak isteyen kullanıcıyı desteklemenin amaçlandığı görsel analitiğin asıl özelliğinin veri keşfi olduğu söylenebilmektedir (Cook ve Thomas, 2005).

Mekânsal görsel analitik, görsel analitiğin bir alt araştırma alanıdır. Analitik düşünme ve karar verme sürecini, hesaplamalı yöntemlerle birbirine bağlantılı etkileşimli görsel ara yüzler aracılığıyla (haritalar ve diğer araçlar gibi) destekleyen, gelişmekte olan çok disiplinli bir alandır (Andrienko ve ark., 2007). Ortaya çıkmasında, karar verme süreçlerini desteklemek için dinamik ve etkileşimli haritaları kullanan mekânsal görselleştirme araştırmaları önemli bir rol oynamıştır (Zhang ve ark., 2015). Mekânsal görsel analitik coğrafi bilgi bilimi, coğrafya, kartografya ve ilgili benzer alanlarda dikkat çeken önemli bir araştırma alanıdır (MacEachren ve ark., 2004;

Andrienko ve ark., 2007). Genel görsel analitik tekniklerinden, insanların bilişsel yetilerini kolaylaştırmak için büyük ölçüde kartografik benzetimlere dayanması açısından, farklılık göstermektedir. Bu nedenle tekrarlı hipotez oluşturma ve doğrulama işlemlerini kolaylaştırmak için bütünleşik ve etkileşimli, harita merkezli analitik ortamlara ihtiyaç duyulmaktadır. Bu hedef doğrultusunda, yenilikçi mekânsal görselleştirme teknikleri ve yüksek performanslı mekânsal analiz araçlarının birleştirilmesi yeni olanaklardır (Zhang ve ark., 2015).

2.4 Önceki Çalışmalar

Ortaya atıldıkları günden itibaren mekânsal görselleştirme, mekânsal görsel analitik, web haritalama servisleri ve web-tabanlı CBS konuları, mekânsal veri ile uğraşan araştırmacıların ilgi odağı olmuştur. Zaten görselleştirme ve mekânsal veri işleme ayrı düşünülemeyecek araştırma alanları olup, mekânsal görselleştirme her zaman mekânsal veriden bilgi üretiminde bütünleyici bir parça rolündedir (Nittel ve ark., 2015). Elbette ki ilgilenilen verinin daha iyi anlaşılabilmesi için farklı görselleştirme teknikleri ve istatistiksel analizlerin bir arada kullanılmasını gerektirecek uygulamalarla da karşılaşılabilir. Görsel analitik bu aşamada devreye girmektedir. Ancak bu yaklaşım daha çok iş dünyasında büyük boyutlu veri problemi ile başa çıkmak için kullanılmaktadır. Mekânsal görsel analitik önceki bölümde de ifade edildiği gibi görsel analitikten türetilmiştir ve konum tabanlı verilerin daha iyi anlaşılabilmesi için yöntemler sunmaktadır. Web haritaları üretim, maliyet, paylaşım ve güncelleme açısından sağladığı avantajlarla her zaman ilgi odağı olmuştur. Gelişen teknoloji bu haritalara gelişmiş fonksiyonellikler eklemeye imkan sunmuş ve web-tabanlı CBS, coğrafi bilgi bilimi alanında trend haline gelmiştir. Bu yaklaşımlar bir araya getirilerek sağlık (Turdukulov ve Moncrieff, 2015), çevre izleme (Marbouti ve ark., 2015), hareket modelleme (Benke ve ark., 2014) ve sosyal medya verisinin analizi (Albuquerque ve ark., 2015) gibi birçok farklı alanda önemli çalışmalar gerçekleştirilmiştir. Bu örnekleri artırmak mümkündür, ancak bu bölümde ele alınan çalışmalar çalışmanın önemi, konu ile ilgi derecesi ve güncelliği göz önünde bulundurularak sınırlandırılmıştır. Çalışmalar sistematik olarak ele alınmıştır. Buradaki amaç, tez çalışması için önemli çıkarımlar yapmak ve anahtar yönlendirmeleri ortaya çıkarmaktır.

Nittel ve ark. (2015) gelecek yirmi yıl içerisinde büyük boyutlu veri probleminin CBS araştırma alanını şekillendireceğini savunmaktadır. Onlara göre teknolojide

yaşanan gelişmeler anlık veri toplayan alıcıların artmasına, otomatikleştirilmiş veri toplama ve analizi işlemlerinin yaygınlaşmasına, hatta günümüzde dahi araştırmaları yapılan, insanların birer alıcıya[19] ya da veri toplayıcıya dönüşmesine sebep olacaktır. Büyük boyutlu veri düşüncesi mekânsal veri ile uğraşan toplulukların genel bir kanısıdır. Bu kanıya değinerek başlamanın doğru olacağı düşünülmüştür. Çünkü teknolojinin elvermediği ya da çok maliyetli olduğu zamanlara nazaran veri ve veriye ulaşma olanaklarındaki zorluklar yerini aşırı miktarda toplanan veri ve bu veriden anlamlı bilgilerin çıkarılması problemine bırakmıştır. Veriye ulaşma yollarındaki kolaylıklar Gelişen web teknolojileri de bu problemde önemli bir rol üstlenmektedir. Özellikle web 2.0 ve HTML5 güncellemelerinin ardından çevrimiçi haritalarla mekânsal-zamansal veri görselleştirme olanakları oldukça gelişmiştir. Artık birbirine bir ağ vasıtasıyla bağlanarak iletişim kuran cihazlar arasında anlık veri transferi ile elde edilen veri/bilgiler anlık olarak görselleştirilip paylaşılabilecek duruma gelmiştir. Sunulan güncelleştirmelerin web haritalama faaliyetlerini nasıl etkileyeceği konusunda test edilmesi, web harita üreticilerini ne gibi yeniliklerin beklediğinin ortaya çıkarılması gerekmektedir. Çünkü bu güncelleştirmeler genel web kullanım trendleri göz önünde bulundurularak gerçekleştirilmekte ve web haritalama alanının bu güncelleştirmelere entegre edilmesi gerekmektedir. 2012 yılında HTML5 henüz beta[20] aşamasındayken Mao ve ark. (2012) yapılan geliştirmeleri, mekânsal-zamansal verilerin görselleştirilmesi için test etmiştir. Onlara göre zamana bağlı verinin iletimi için WebSocket[21] protokolü, etkili dinamik görselleştirme için de canvas[22] elemanı özellikleri uygun bir şekilde kullanılabilmektedir. Uygulamalarını Google harita servisi ile gerçekleştiren araştırmacılar oluşturulan konum tabanlı uygulamanın etkili ve kullanımı kolay olduğunu belirtmişlerdir. Görselleştirmenin daha etkili yapılabilmesi için zamansal verinin genelleştirilmesinin, gelecekte üzerinde durulması gereken bir konu olduğunu ve mekânsal-zamansal verilerin 3 boyutlu görselleştirilmesi amacıyla HTML5'in desteklediği WebGL[23], X3DOM[24] gibi üç boyutlu teknolojilerin

[19] Nittel ve ark., konum duyarlı teknolojiler ve akıllı telefonlar (GPS alıcısına sahip) sayesinde insanların her an her yerde konuma dayalı veri toplayıp paylaşabildiğini, bu sayede de insanların birer alıcı ya da veri toplayıcı olarak nitelendirilebileceğini savunmaktadır.
[20] HTML5 web üzerinde içeriklerin yapılandırılması ve sunulması için kullanılan bir işaretleme dilidir.W3C konsorsiyumu tarafından 28 Ekim 2014 yılında tamamlanmış ve yayınlanmıştır.
[21] WebSocket protokolü tek bir TCP bağlantısı üzerinden tam çift yönlü iletişim kanalları sağlayan bir protokoldür.
[22] HTML <canvas> elemanı bir web sayfasına grafik çizmek için kullanılmaktadır.
[23] WebGL (Web Graphics Library) bir JavaScript API olup etkileşimli 3 ve 2 boyutlu bilgisayar grafiklerinin işlenmesi ve herhangi bir eklentiye ihtiyaç olmadan web tarayıcılarında kullanılabilmesini sağlayan bir teknolojidir.

araştırılması gerektiğini savunmaktadırlar. Aslında adı geçen teknolojilerin tamamı uyarlamalı teknolojiler olduğundan, gerçekleştirilen çalışmalar web haritalama işlemleri ve web-tabanlı CBS sistemlerinin geleceğini şekillendirmektedir. Bu çalışmalara verilebilecek en iyi örneklerden biri de sağlık alanında gerçekleştirilen ve mekânsal görsel analitik ile GoogleEarth yazılımının hastalık dağılımının daha iyi anlaşılabilmesi için gerçekleştirilen çalışmadır. Marek ve ark. (2015) Google Earth™ yazılımının bir mekânsal görsel analitik aracı olarak kullanılabilirliğini hastalık verisi üzerinde test etmiştir. Elde ettikleri sonuçlar incelendiğinde GoogleEarth yazılımının mekânsal görsel analitik yaklaşımıyla hastalık dağılımlarının araştırılmasında kullanışlı bir araç olduğu iddia edilmektedir. Bunun sebebini ise sahip olduğu geniş çaplı kullanım, ücretsiz erişim, sezgisel ara yüz, zaman-mekân görselleştirme yetenekleri, animasyonlar ve sonuçların paylaşımı gibi tartışmasız birçok avantajına bağlamaktadırlar. Bununla birlikte, halen, verinin mekânsal görsel analitiğe uygun forma getirilmesi açısından veri ön işleme araçlarıyla birleştirilmesi gerektiği de belirtilmektedir. Aslında bu durumun sebebi, bu gibi yazılım ve servislerin ortaya çıkış amacıdır. Başlarda sadece konum paylaşma, yer etiketleme ya da sosyal paylaşımların konumla ilişkilendirilmesi için ortaya çıkan bu teknolojiler kullanıcı gereksinimleri ile günümüze kadar şekillendirilmiştir.

Anlık toplanabilen ve büyük boyutlara sahip veri problemini, gelişmiş web-tabanlı uygulamalarla bir fırsata dönüştürme potansiyeline sahip uygulamalara ulaşmak da mümkündür. Örneğin sosyal medyada her gün paylaşılan ve içerisinde konum bilgisi de ihtiva eden yüzbinlerce mesajdan anlamlı bilgi çıkarımı son zamanlarda oldukça dikkat çeken bir araştırma alanıdır. Albuquerque ve ark. (2015) Almanya'da bir sel felaketi esnasında paylaşılan ve konum bilgisi de içeren tweetlerle (içerisinde felaketle ilgili anahtar kelimeler barındıran) güvenilir verileri (alıcı verisi, hidrolojik veri ve sayısal yükseklik modeli gibi) birlikte analiz ederek, selden etkilenen bölgelerin tespit edilmesinde kullanmış ve bu gibi uygulamaların afet yönetimi ve erken uyarı sistemlerinde kullanılabileceği potansiyelini ortaya çıkarmıştır. Benzer bir uygulamada Nelson ve ark. (2015) Amerika Birleşik Devletleri'nde yeni yürürlüğe giren ve halkı düşünce açısından ikiye bölen bir uygulamanın twitter mesajları üzerinden demografik dağılımını ortaya konmaya çalışılmıştır. Sosyal medya ve mekânsal verilerin birleştirildiği web-tabanlı çalışmaların daha birçok örneğine ulaşmak mümkündür örn.,

[24] X3DOM, WebGL aracılığıyla işlenen 3 boyutlu görüntülerin HTML5 DOM elemanları gibi entegrasyonu ve değiştirilmesi için geliştirilen bir teknolojidir.

bkz. (Sukic ve Stoimenov, 2012; Ghani ve ark., 2013; Luo ve MacEachren, 2014; Turnbull ve Randhawa, 2015).

Üçüncü bölümde daha detaylı olarak ele alınacak mashup kavramının web ortamında farklı servisleri bir araya getirebilme yeteneği olduğu önceki bölümlerde ifade edilmiştir. Böylesine bir avantaj web harita üretiminde de güçlü uygulamaların oluşturulmasına imkân sunmaktadır. Web üzerinde farklı web sitelerinden alınan servislerin bir araya toplamak, değiştirmek ve mashup uygulamaları oluşturmak için geliştirilen Yahoo Pipes[25], Microsoft Popfly[26], Google Mashup Editor[27], ve IBM Mashups[28] benzeri derleme araçlarına kısa bir süre öncesine kadar ulaşmak mümkündü. Bu araçlar, zamanla bütçe sıkıntıları ya da diğer hizmetlerine daha iyi odaklanma sebepleriyle şirketler tarafından kapatılmıştır. Han ve ark. (2013) bu araçların kullanılabildiği zamanlarda dahi farklı türdeki web içerikleriyle sağlanan veri, hesaplamalar ve kullanıcı ara yüzlerine erişmek için tek bir ara yüzün mevcut olmadığını ifade etmektedir. Onlara göre mashup uygulamalarının çoğunun beslendiği web içerikleri, web servis API'leri aracılığıyla üçüncü parti kaynaklardan elde edilmektedir. Bu, artık mashup uygulamalarının oluşturulmasında, kullanıcıların kendi mimarisini ve yapısını oluşturma zorunda kalacağı anlamına gelmektedir.

Birçok farklı alanda gerçekleştirilen web-tabanlı CBS çalışmaları incelendiğinde (Singh ve ark., 2012; Werts ve ark., 2012a; Al-Ahmadi ve ark., 2013; May ve ark., 2014; Misra ve ark., 2014; Parker ve ark., 2014) odaklanılan asıl konunun ilgili verinin halihazırda CBS fonksiyonelliklerine sahip uygulamaların (örn., ArcGIS server gibi) API ara yüzleri kullanılarak web üzerinden dağıtımının yapılabilmesi olduğu anlaşılmaktadır. Bu çalışmaların hiç biri mevcut ücretsiz web servislerine CBS fonksiyonellikleri kazandırmak ya da yeni araçlarla mekânsal görsel analitik ortamı oluşturmak konularına odaklanmamıştır. Bu konular ve incelenen çalışmalardan elde edilen kazanımlar sonucunda, aşağıda sıralanan çıkarımlar ve yönlendirmelerle tez çalışması şekillendirilmeye çalışılmıştır.

- Web alanında sunulan yeni teknolojiler, mekânsal-zamansal verinin etkileşimli görselleştirilmesi açısından birçok olanak sunmaktadır (HTML5, AJAX, REST, SOAP, WebGL, Canvas vb.).

[25] 2015 itibari ile servis Yahoo tarafından durdurulmuştur.
[26] 2009 itibari ile servis Microsoft tarafından durdurulmuştur.
[27] 2009 itibari ile servis Google tarafından durdurulmuştur.
[28] 2012 itibari ile servis IBM tarafından durdurulmuştur.

- Birçok web haritalama servisi mevcut olup sundukları olanaklar bakımından farklılık göstermektedir (desteklenen veri formatı, gizlilik politikası, harita özellikleri vb.).
- Web-tabanlı CBS benzeri mashup uygulamaları, uyarlamalı[29] uygulamalar olup gelişimlerini bu alanda gerçekleştirilecek çalışmalarla sürdürebilecektir (web haritalama servislerinin genellikle (ücretsiz olanlar) CBS fonksiyonelliklerine sahip olmaması ve bu özelliklerin API ve Mashup teknolojileri ile kullanıcılar tarafından dahil edilmesi).
- Harita mashup uygulamalarının oluşturulabileceği bir araç günümüzde mevcut olmadığı için, oluşturulan uygulamadaki entegre servisler ve harita araçları, teknolojik olanaklara ve oluşturucusunun hayal gücüne bağlıdır (HTML, CSS ve JavaScript dilleri kullanılarak oluşturulan mashup uygulamaları programlama düzeylerine göre gelişmişlik düzeyini artırma potansiyeline sahiptir).
- Mevcut web haritalama servislerinin veri ön işleme süreçleri yetersizdir ve üçüncü parti araçlarla desteklenmelidir (özellikle verinin analizi konusunda web haritalama servisleri yetersiz kalmaktadır, bu nedenle verinin üçüncü parti yazılımlarla ön işleme tabi tutulup analiz edilerek sonuçların web haritaları üzerinde dinamik gösterimleri tercih edilmelidir).
- Gelecekte web haritalama faaliyetlerini şekillendirecek en önemli konu büyük boyutlu verilerin (alıcılar, kitle-kaynaklı veri ve sosyal medya ağlarından toplanan veri gibi) işlenmesidir (gerçekleştirilen bu tez çalışmasında ve çalışmanın gelecek adımlarında bu konunun da değerlendirilmesi gerekmektedir)

[29] Uyarlamalı uygulama kavramı, uygulamaların gelişmişlik derecesi mevcut teknolojilerin özel bir alana entegrasyonu anlamında kullanılmıştır.

3 ÇEVRİMİÇİ HARİTALAR: İLGİLİ TEKNOLOJİLER, MASHUP ve API KAVRAMLARI

Çevrimiçi harita kavramı, web üzerinden çevrimiçi olarak erişilebilen haritaları temsil etmektedir. 1992 yılında ortaya çıkışından itibaren, haritalama alanında gerçekleştirilen çalışmalar kısa sürede web ortamında adapte olmuştur. Hemen ardından 1993 yılında ilk statik harita sunucularından biri olan Xerox PARC Map Viewer ortaya çıkmıştır (Putz, 1994). Web haritaları ortaya çıkışının ardından kartografya alanında yeni bir eğilim haline gelmiştir. Kısa bir süre öncesine kadar haritacılık faaliyetleri maliyetli ve karmaşık yazılım ve donanım ile deneyimli kartograflar gereksinimleriyle, birkaç şirketin, enstitünün ya da askeri kuruluşların imkânlarıyla sınırlıydı. Web haritalama kavramının ortaya çıkışı ile birçok yeni teknoloji ve veri de gündeme gelerek bu sınırı bulanıklaştırmıştır. Web üzerinde kullanıcıların erişebileceği kitle-kaynaklı veri sağlayan OSM ya da Google, Microsoft, Yahoo! gibi ticari şirketler tarafından sağlanan verilere ulaşmak mümkündür. Bunun yanında bu veriler kullanılarak web haritalarının oluşturulabileceği birçok ücretsiz yazılım da ArcGIS gibi ticari benzerlerinin yanında yerini almıştır. Sonuç olarak web üzerinde harita üretmek ve yayınlamak artık oldukça kolaylaşmıştır.

Geleneksel harita üretim teknikleri ile karşılaştırıldığında, web haritalarının aşağıda sıralandığı gibi birçok avantajını saymak mümkündür:

- Web haritaları güncel veri yayınlamak için oldukça elverişlidir. Veri tabanlarına bağlanarak üretilen web haritaları, veriyi anlık olarak gösterebilme avantajına sahiptir. Basılmaları, çoğaltılmaları ya da yayımlanmalarına gerek yoktur.
- Web haritaları için yazılım ve donanım maliyetleri oldukça düşüktür. Web sunucu donanımları düşük maliyetlerle elde edilebilmekte ve web haritalarının üretiminde birçok açık kaynak yazılıma ulaşılabilmektedir.
- Ürün güncelleştirmeleri kolaylıkla yapılabilmektedir. Yapılan güncelleştirmelerin dağıtımı, web kullanıcılarının web sayfasını yeniden yüklemesinden ibarettir.
- Web haritaları, web üzerinden dağıtımı yapılan farklı veri kaynaklarını bir araya getirebilmektedir. API ara yüzleri kullanılarak oluşturulan haritalar

(mashup) projeksiyon, harita ölçeği ve veri kalitesi uygun olduğu taktirde farklı veri kaynaklarını bir arada sunabilmektedir.

- Web haritaları kişiselleştirilebilmektedir. Web haritalama sistemleri el verdiği müddetçe, kullanıcı profilleri, kişisel filtreler, kişisel tasarım ve işaretleştirmelerle, kullanıcılar kendi özel haritalarını üretebilmektedir.
- Web haritaları, kullanıcıların oluşturduğu verilerin bir araya getirilerek üretildiği haritalar gibi, ortak harita üretimine olanak sağlamaktadır. OSM ya da Google Earth™ toplulukları gibi örnekleri çoğaltmak mümkündür.
- Web haritaları, harita elemanlarıyla web üzerinde yayınlanan diğer çoklu ortamların birbiri ile ilişkilendirmesine izin vermektedir. Yani bir harita elemanına, web üzerinde bulunan herhangi bir bilgi, resim ya da video gibi çoklu ortam elemanları bağlantılanabilmekte, bu sayede çok daha zengin içerikli haritalar oluşturulabilmektedir.

Sözü edilen bu avantajların yanında, web haritalarının bazı sakıncaları da elbette mevcuttur. Bunlara ilk olarak mekânsal verinin dağıtım maliyetlerinin, web haritaları kadar düşük olmaması verilebilmektedir. Uydu ya da araçlara takılan donanımlar sayesinde toplanan mekânsal veriler oldukça yüksek maliyetlerle elde edilmektedir. Kullanıcılar büyük şirketlerin ya da diğer kaynakların (örn., kurumlar) yayınladığı mekânsal verileri kullanırken her zaman telif hakkı ya da kullanım izinleri ile uğraşmak zorunda kalabilmektedir. Bir diğer konuda web haritalarının güncelleme kolaylığından dolayı çoğu zaman işaretleştirme ya da doğruluk bakımından düşük kaliteli ürünler ortaya çıkmaktadır. Bunun nedeni güncellemelerinin ucuz, hızlı ve kolay yapılabilmesidir. Geleneksel kartografyada ise basılı haritalar ya da bir ortamda (CD, DVD) dağıtımı yapılan haritaların güncellemeleri söz konusu olduğunda, haritanın yeniden basımı, çoğaltılması ve kontrolü gibi birçok çaba gerektiren işlem adımından söz edilebilmektedir. Düşük doğruluk kavramı, faklı kaynaklardan bir araya getirilen verilerle üretilen web haritalarında daha çok ön plana çıkmaktadır. Çünkü kullanıcılar çoğu kez dış kaynaklardan alınan verilere güvenmek zorunda kalmaktadır.

Yukarıda bir avantaj olarak bahsedilen web haritalarının kişiselleştirilebilme yetenekleri bazı durumlarda olumsuz sonuçların ortaya çıkmasına yol açabilmektedir. Buna bir örnek olarak, deneyimsiz ya da uzman olmayan kullanıcıların oluşturduğu haritalarda kullanılan uygunsuz renk seçimleri, renk körleri için haritanın okunamaz ya da yanlış anlaşılabilir olmasına sebep olabilmektedir. Bu dezavantajlara ek olarak baskı haritalara benzer şekilde web haritalarında da sınırlı gösterim alanı problemi

yaşanmaktadır. Özellikle mobil cihazların küçük ekranlarında harita okunabilirliği oldukça düşebilmektedir.

Her ne kadar web haritalarının üretimi ve maliyetleri kolay ve ulaşılabilir olsa da, oluşturulabilmeleri için ücretsiz ya da ticari birçok alternatif gün geçtikçe artsa da, etkileşimli, farklı web kaynaklarını bir araya getiren karmaşık web haritaları ya da web CBS uygulamaları oluşturmak günümüzde dahi zor bir görevdir. Bunun için birçok teknoloji, modül, servis ve veri kaynağı hakkında deneyim sahibi olunması gerekmektedir. Farklı web teknolojilerinin geliştirme ve hata ayıklama ortamları halen kolay anlaşılamaz ve karmaşıktır. Takip eden bölümlerde web haritaları oluşturma aşamasında ihtiyaç duyulan teknolojiler, web haritalama servisleri, API ve mashup kavramları detaylı olarak açıklanmaktadır. Ardından bir web CBS uygulaması için kullanılacak uygun servislerin belirlenen kıstaslar doğrultusunda değerlendirilmesi amaçlanmıştır.

3.1 İlgili Teknolojiler

Güncel web haritalama teknolojileri ve web teknolojileri Roth ve ark. (2015) tarafından da ifade edildiği üzere üç ana bölüme ayrılabilmektedir. İlk olarak merkezi ya da giderek artan şekliyle dağıtımı yapılan bir kaynaktan (örn., bulut tabanlı kaynak) coğrafi bilgilerin dizinlendiği ya da sorgulandığı sunucu-taraflı teknolojilerdir. İkinci olarak kullanıcıların tarayıcılarında, bu coğrafi bilgilerin işlenmesi ve değiştirilmesi ile web haritalarının oluşturulduğu, istemci-taraflı teknolojiler bulunmaktadır. Son grupta da istemci ve sunucu arasında bilgi isteklerinin aktarımı için kullanılan web servisleri ya da diğer aracı makrolar bulunmaktadır (Roth ve ark., 2008). Aslında Roth ve ark. (2015) iş web haritalarına geldiğinde, bir kartografla bir CBS teknisyeni arasındaki farkın, istemci uygulamalarıyla ortaya çıktığını savunmaktadırlar. Çünkü, haritanın sunumu (projeksiyon seçimi, genelleştirme, işaretleştirme vb.), bu haritanın farklı amaçlar için uyarlanmasında kullanılacak kullanıcı ara yüzü ve web haritası ile elde edilen genel deneyim için tasarım aşamasındaki karar alma işlemlerinde, kartograflar tarafından, istemci uygulamalarıyla yapılan katkı daha fazladır. Bu nedenle burada da istemci-tarafı teknolojilere odaklanılmıştır. Web haritalama alanında istemci-tarafı teknolojiler HTML, XML (eXtensible Markup Language) , SVG (Scalable Vector Graphics), CSS

ve DOM gibi web standartlarını içermektedir. Bu standartlar W3C[30] (World Wide Web Consortium, W3C) tarafından belirlenmektedir.

HTML web dokümanlarını (web sayfalarını) tanımlamak için kullanılan bir işaretleme (sayfa tanımlama) dilidir. Bir işaretleme dili, bir dizi işaretleme etiketinden oluşan dili temsil etmektedir. HTML dokümanları bu etiketlerle tanımlanmakta her bir etiket farklı bir web sayfası içeriğini temsil etmektedir. Aslında bir web sayfasının oluşturulması için gereken ana teknolojidir. Diğer teknolojiler, bir web sayfasının oluşturulmasında HTML yapısının içerisine dâhil edilmektedir. En temel anlamda web geliştiricilerinin mutlaka bilmesi gereken üç ana bileşenden biridir. Diğer iki bileşen, web sayfasının tasarımının ve web sayfasındaki nesnelerin tasarım açısından nasıl görüneceğini belirleyen CSS ve web sayfasının ve içerdiği nesnelerin davranışlarını kontrol eden JavaScript dilleridir.

Daha gelişmiş ve amaca uygun özel web uygulamaları oluşturmada, diğer teknolojiler bu üç ana bileşene entegre edilmektedir. Örneğin XML, genellikle web üzerinde veri formatı olarak ya da veri transferi için kullanılan ve HTML dilinden daha esnek bir yapıya sahip işaretleme dilidir. ESRI (Environmental Systems Research Institute) tarafından kullanılan ArcXML ya da daha bilindik bir örnekle Google tarafından kullanılan KML (Keyhole Markup Language), XML dili ile oluşturulan veri formatlarına verilebilecek örnekler arasındadır. SVG olarak anılan standart ise farklı türde bir XML olup web tarayıcılarında vektörel grafikler ve metinlerin tanımlanması ve işlenmesine imkân veren bir teknolojidir. Web tarayıcısı içerisinde HTML ve XML objelerinin oluşturulması ve amaca uygun olarak uyarlanması için kullanılan ve platform-bağımsız bir tanımlama olan DOM, JavaScript kullanılarak tarayıcı içerisindeki objelerin uyarlanabilmesini sağlamaktadır. JavaScript dilinin web geliştirme alanında temel üç dilden biri olduğu daha önceden ifade edilmiştir. Bunun sebebi tüm modern web tarayıcıları tarafından herhangi bir eklenti gerektirmeden destekleniyor olmasıdır.

Bu teknolojiler kullanılarak web haritalama, CBS ve konum tabanlı servislerdeki gelişmişlik düzeyinin artması, mekânsal verilere ulaşım, kullanım ve işlemeyi daha önce olmadığı kadar ileri götürmüştür. Bu işlemde farklı teknolojiler arası iletişimin gelişmesi ve büyük boyutlu mekânsal verilerin paylaşımı önemli bir rol oynamaktadır. Google, Microsoft ve Yahoo! gibi şirketlerin bu gelişmelerde rolü büyüktür. 2005

[30]World Wide Web ya da kısaca Web yada genel kullanımda W3C tarafından belirlenen standartlara http://www.w3.org/ adresinden ulaşılabilmektedir

yılında Google Maps ile AJAX'ın, sunucular ve istemciler arasındaki ilişkiler için yaptıkları birleşim büyük bir yeniliktir. Bu yenilik Internet aracılığıyla gerçekleştirilen iletişimde doruk noktası olarak sayılabilmektedir (Peterson, 2012a). Temelde AJAX, kullanıcının özel bir sorgu gerçekleştirmediği sürece, arka planda sunucu tarafından küçük mesajların işlendiği sürekli bir bağlantı sağlamaktadır (Garrett, 2005). Bu sayede, kullanıcı sorgu gerçekleştirdiğinde, daha hızlı sunucu cevapları alınabilmektedir. AJAX aslında kullanıcının ne gibi sorgular gerçekleştirebileceğini bekleyen, web tarayıcısı içinde bulunan bir arka plan uygulaması gibi düşünülebilir. Google Maps üzerinden gerçekleştirilen işlemler genel olarak AJAX desteği içermektedir ve bu işlemler görüntü yakınlaştırma/uzaklaştırma, kaydırma ya da en genel şekliyle harita ile etkileşim işlemlerini içermektedir. AJAX kendi içerisinde bir programlama dili değildir ancak bir grup farklı teknolojinin birleşimini temsil etmektedir. Aslında HTML, CSS, DOM ve XML teknolojilerinin bir karışımıdır. Bu teknolojilerin tamamına ücretsiz olarak erişilebilmektedir. Teknolojinin ilk harfini temsil eden "A" harfi eş zamansız iletişimi temsil etmekte olup sunucu ile veri değişiminin, kullanıcı web sayfasını görüntülediği esnada yapıldığını ifade etmektedir. Bu, kullanıcının sayfa içerisindeki bir bölümde değişen içeriğin yüklenmesi için tüm sayfanın yeniden yüklenme zorunluluğunu ortadan kaldırmaktadır. Sonuç olarak daha iyi etkileşim düzeyi, hız ve daha gelişmiş ara yüzler geliştirilebilmektedir.

Web'in ilk ortaya çıktığı zamanlardan günümüze, web üzerinden görüntü dosyalarının dağıtımı için mozaik[31] mantığı kullanılmaktadır. Metinlere nazaran görüntüler her zaman daha büyük boyutlu depolama alanları ve daha uzun yükleme zamanlarına ihtiyaç duymaktadır. Bir çözüm olarak web üzerinde görüntülerin dağıtımı yapılırken mozaik adı verilen küçük parçalara bölünmekte, bu sayede sadece görüntülenen alan için görüntü parçaları ayrı ayrı yüklenerek ve kullanıcı bilgisayarına farklı yollar izleyerek eriştiği için daha hızlı transferler gerçekleştirilmektedir. Web haritalarında da sıklıkla bu yöntem kullanılmakta, farklı yakınlık derecelerine göre sunuculardan faklı görüntüler çağırılabilmektedir. Yeterince hızlı bir internet bağlantısı ile bu işlem oldukça kısa sürelerde gerçekleşebilmekte ve kullanıcı çoğu zaman görüntülediği haritanın farklı mozaik parçalarından oluştuğunu hissetmemektedir. Aynı durum harita üzerine, harita yüklendikten sonra herhangi bir katman ya da nesne ekleme işleminde de geçerlidir. Sözü edilen AJAX teknolojisi sayesinde kullanıcılar harita

[31] Bu çalışmada, kökeni "map tile" olarak ortaya çıkmış kavram mozaik olarak anılmaktadır.

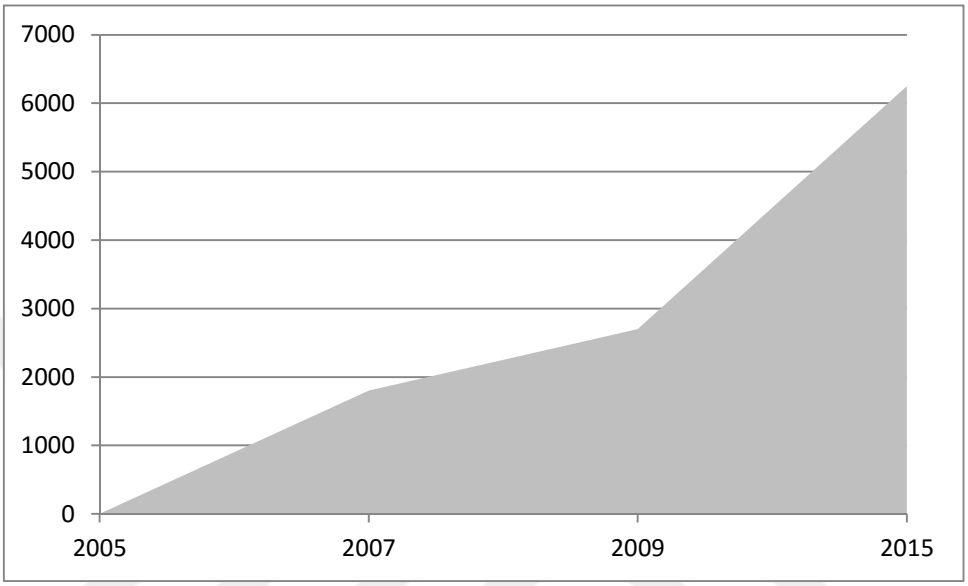
üzerinde gezdiği ya da görüntüyü yakınlaştırdığı zaman, faklı harita mozaikleri sayfa yeniden yüklenmeden sayfaya eklenebilmekte, bu şekilde akıcı bir kullanım gerçekleştirilebilmektedir.

Bu teknolojilerin yanında çevrimiçi coğrafi bilgi sistemleri veri tabanlarından mekânsal verileri çekebilmek için kullanılan Web Map Service (WMS), Web Feature Service (WFS) ve Web Coverage Service (WCS) gibi standartlar da Open Geospatial Consortium (OGC) tarafından tanımlanmıştır. Amaç, çevrimiçi servislerde tutulan farklı boyutlarda ve farklı formatlardaki coğrafi verilerin, birçok kullanıcı tarafından kullanılabilmesi için bir veri dağıtım standardının oluşturulmasıdır. WFS, mekânsal referanslı ve genellikle raster mozaik formatında (PNG, GIF ya da JPG) ya da bazen vektör formatta harita görüntülerini temsil etmektedir. İstekler standart web URL adresleri ile gerçekleştirilmektedir. WCS, bir harita üzerinde görüntü bindirme yöntemi ile eklenebilen bir coğrafi alanı temsil etmektedir. Ancak bu alan üzerinde herhangi bir değişiklik yapılamaz ya da analiz gerçekleştirilemez. WCS, noktalar, pikseller ya da vektörlerle tanımlanan güzergâh nesnelerinden oluşan kapsamların transferi için kullanılmaktadır. WFS ise coğrafi elemanların çağırılması için kullanılmaktadır. WFS, kullanıcılar tarafından elemanların sorgulanması, değiştirilmesi, güncellenmesi ya da silinmesine izin vermektedir. Veri genellikle GML (Geography Markup Language) gibi XML formatında sağlanmaktadır.

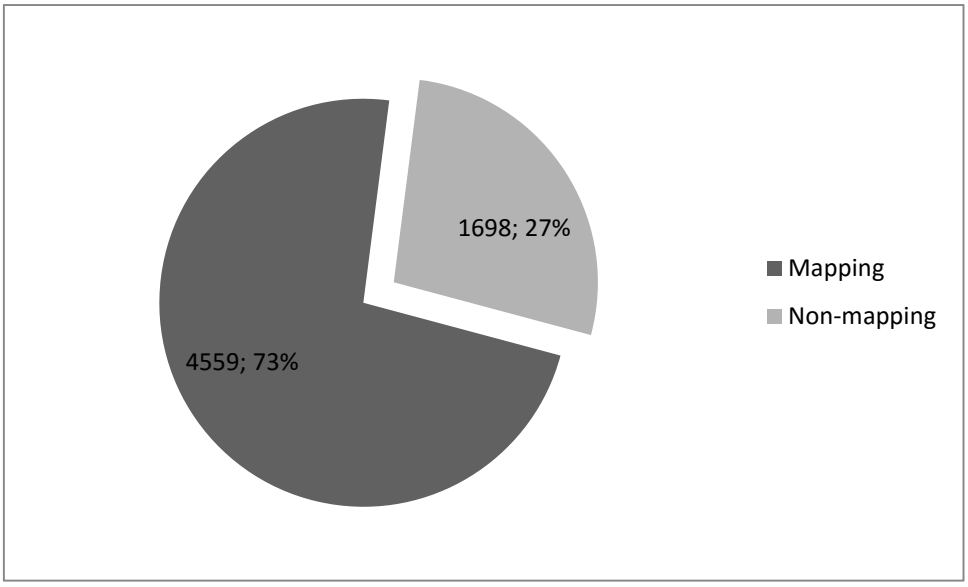
3.2 Mashup Kavramı

Web teknolojilerinde yaşanan gelişmeler “Web Mashup” teriminin ortaya çıkmasına zemin hazırlamıştır. “Mashup” aslında bir müzik terimi olarak ortaya çıkmıştır. Farklı müzik parçalarının bir araya getirilerek kesintisiz olarak birleştirilmesi ya da karıştırılması ile oluşan yeni parçayı temsil etmektedir. Aynı durum video parçaları için de geçerlidir. Önceleri müzik ve video alanlarında kullanılan bu terim günümüzde Web dünyasında yaşanan gelişmelere de bağlı olarak ikinci kuşak web teknolojisinin (Web 2.0) giderek yaygınlaşan uygulamalarından biri olarak yerini almıştır. Web alanında bu terim birden fazla kaynaktan alınan içeriklerin, yeni bir hizmet oluşturmak amacıyla, tek bir grafik ara yüzde birleştirilmesi, anlamında kullanılmaktadır. Daha önce de belirtildiği gibi internet üzerinde toplam API ve mashup sayısının resmi olarak belirlenmesi mümkün değildir. Ancak bu gibi uygulamaların gün geçtikçe arttığı ortadadır (Şekil 3.1). Bu sayıya ulaşmak için Weiss ve Gangadharan

(2010), Chow (2011) ve Programmableweb (2016) kaynaklarından yararlanılmıştır. Çoklu veri kaynaklarının birleştirilmesi ve elde edilen sonuçları web üzerinden yayınlanması nedeniyle mashuplar uygulama-odaklıdır. Çok sayıda mashup uygulaması oluşturulmasına rağmen, bu uygulamaların büyük bir bölümü harita bağlantılı uygulamalardır (Şekil 3.2).



Şekil 3.1 İnternette ilk görülmeye başladığı andan günümüze mashup sayısının zaman içerisindeki artışı (Programmableweb, 2016)



Şekil 3.2 Harita bağlantılı mashup uygulamalarının tüm mashup uygulamalarına oranı (Programmableweb, 2016)

Birincil işlevinin çoklu veri kaynaklarını bir araya getirmek olduğu düşünüldüğünde, mashup uygulamalarında en önemli bileşenlerden biri veridir. Birçok

mekânsal veri sağlayıcısının konuma dayalı veri yayınlamasına rağmen (örn. trafik verisi) çoğu zaman özel verilerin mashup uygulamasına eklenmesi istenmektedir. Oldukça çok sayıda ve farklı şekillerde tanımlanan web veri formatları bulunmasına rağmen burada mekânsal ya da haritalama kaynaklı web veri formatları detaylandırılmıştır.

3.2.1 Mekânsal veriler için bazı yaygın veri formatları

3.2.1.1 GML

GML, coğrafi nesneleri konumsal ve öznitelik bilgileri ile birlikte kodlamak için kullanılan bir XML uzantısıdır. GML 3.1 basit geometrileri (nokta, çizgi, alan), karmaşık geometrileri (basit geometrilerin bir araya gelerek oluşturduğu kapalı objeleri), geometrik kümeleri (çoklu nokta, çoklu doğru, çoklu çokgen, çoklu geometri gibi) ve raster görüntüleri (grid dosyaları) desteklemektedir. Şekil 3.3'de bir GML kodlamasında geometrilerin ifade ediliş şekilleri örneklenmektedir. CityGML 3 boyutlu objeleri sanal bir kent oluşturmak için kodlayan GML dilini temsil etmektedir.

```
<gml:Polygon>
         <gml:outerBoundaryIs>
                 <gml:LinearRing>
                         <gml:coordinates>0,0 100,0 100,100 0,100
0,0</gml:coordinates>
                 </gml:LinearRing>
        </gml:outerBoundaryIs>
     </gml:Polygon>
     <gml:Point>
        <gml:coordinates>100,200</gml:coordinates>
     </gml:Point>
     <gml:LineString>
        <gml:coordinates>100,200 150,300</gml:coordinates>
     </gml:LineString>
```

Şekil 3.3 GML kodlamasında geometrilerin ifade ediliş şekli

Bu nedenle GML gerçek dünya nesnelerini göstermek için çok yönlü veri modelleri ve geometrileri kapsamaktadır. Chow (2008) GML verisinin Google Maps üzerinde görüntülenebileceğini örneklemek için gerçekleştirdiği çalışmada, bu gibi açık tanım formatların web CBS çözümlerinin geliştirilmesindeki potansiyelini ortaya koymaktadır. Birçok vektör ve raster veri formatını GML formatına dönüştüren birçok

ücretsiz uygulamaya (mygeodata vb.) ve ticari uygulamaya (ArcGIS) erişmek mümkündür.

3.2.1.2 KML

KML, GML'e benzer şekilde coğrafi nesneleri tanımlamak için belirli bir XML şeması kullanmaktadır (Şekil 3.4) (GoogleDevelopers, 2016c). İki işaretleme dili arasındaki en büyük fark KML formatında görselleştirme ve navigasyon parametrelerinin de (<Camera>, <LookAt>, <gx:FlyTo>) yer alabilmesidir. KML formatı sadece basit geometrileri desteklemektedir. Bunlar arasında marker (nokta), çoklu doğru, kapalı çokgen ve raster mozaikler bulunmaktadır. Bu nedenle gerçek dünya nesnelerinin ve bu nesnelerin mekânsal ilişkilerinin görüntülenmesinde GML formatına göre oldukça sınırlı kalmaktadır. Buna rağmen Google Earth, Nasa World Wind, Microsoft Bing Maps 3D gibi diğer sanal küreler ve daha birçok web harita servisi KML kodlamasını desteklemektedir ve KML katmanına izin vermektedir.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document>
<Placemark>
  <name>New York City</name>
  <description>New York City</description>
  <Point>
    <coordinates>-74.006393,40.714172,0</coordinates>
  </Point>
</Placemark>
</Document>
</kml>
```

Şekil 3.4 KML kodlamasında bir marker işaretinin ifade ediliş şekli

Open Geospatial Consortium (OGC) KML 2.2 sürümünü 2008 yılında bir uygulama standardı olarak kabul etmiştir. Küresel veri kapsamı (hava/uydu fotoğrafları, vektör tabanlı harita, arazi modeli, StreetView görüntüleri), KML verisinin kolay üretimi ve dağıtımı nedeniyle, afet durumları gibi acil durumların coğrafi verisinin bir an önce büyük topluluklara aktarılması gereken işlemlerde KML her zaman ilk çözüm olarak akıllara gelmektedir.

3.2.1.3 GeoRSS – GeoRSS GML

GeoRSS kelime kökeni olarak Really Simple Syndication (RSS) teriminden gelmektedir. Çok yaygın bir XML tabanlı yayın ve birleştirme formatıdır. Bir internet uygulamasının web içeriği, bir dış sağlayıcı tarafından sürekli güncellenmektedir. Web yayını, orijinal metinlerin biraz daha basit bir şekli olup bir başlık, bir özet, orijinal içeriğe bir bağlantı ve tarih/zaman içeren yardımcı bir meta veriden oluşmaktadır. GeoRSS ifadesindeki "geo" ön eki, RSS yayınları için coğrafi olarak kodlanmış nesneleri temsil etmektedir. RSS dosyası içerisinde ek olarak mekânsal bilgi yerleştirilmektedir. Mekânsal bilgi, genel olarak tam içerikle ilgili konum bilgisini içermektedir. Genellikle bu gibi yayınlar haber akışları, borsa gibi uygulamalarda kullanılır. Bir masaüstü ya da web RSS toplayıcı uygulamasına yayıncının adresi girilerek akışların alınması sağlanır. GeoRSS verisine örnek olarak da USGS (United States Geological Survey) tarafından yayınlanan deprem verisi verilebilir. Bu sayede meydana gelen depremler anlık olarak harita üzerinde görüntülenebilmektedir.

İki tür GeoRSS formatı mevcuttur. Bunlardan birincisi nokta, çizgi, alan ve kutu (sınırlayıcı dörtgen) gibi basit geometrileri destekleyen basit GeoRSS, diğeri de GeoRSS GML kodlamasıdır. Bu kodlamada, nesne türü, nesne ilişkileri, yükseklik, yarıçap gibi öznitelikler, nesnelerin 3 boyutlu uzayda gösterimi için kullanılabilmektedir. GeoRSS GML, basit GeoRSS kodlamasında olduğu gibi basit geometrileri desteklemektedir Ancak koordinatların tanımlanmasında GML uygulama şemasını ve etiketlerini kullanmaktadır (GeoRSS, 2016). Şekil 3.5, basit ve GML GeoRSS kodlamalarıyla bir noktanın nasıl ifade edildiğini örneklemektedir.

```
Simple GeoRSS
<georss:point>45.256 -71.92</georss:point>

GeoRSS GML
<georss:where>
  <gml:Point>
    <gml:pos>45.256 -71.92</gml:pos>
  </gml:Point>
</georss:where>
```

Şekil 3.5 Basit GeoRSS ve GeoRSS GML kodlamalarında bir noktanın gösterilişi

Basit GeoRSS kodlamasında WGS84 (World Geodetic System 1984) datumunda, ondalıklı enlem, boylam çiftleri nokta koordinatlarını temsil etmektedir.

GeoRSS GML kodlamasında ise bunun yanında UTM (Universal Transverse Mercator) gibi diğer koordinat sistemleri de desteklenmektedir. Bu sayede geometrilerin daha iyi temsili ve daha yüksek harita doğruluğu elde edilebilmektedir. Gerçek zamanlı güncellenebilme yeteneği sayesinde GeoRSS, harita mashup uygulamalarının oluşturulmasında uygun bir veri girdi kaynağı olarak kullanılabilme potansiyeline sahiptir. Google Maps, Yahoo! Maps, Microsoft Bing Maps 3D gibi büyük web harita servis sağlayıcıları tarafından desteklenmektedir.

3.2.1.4 GeoJSON - TopoJSON

GeoJSON[32], coğrafi verilerin kodlanması için JavaScript Object Notation (JSON) kullanan basit bir veri değişim formatıdır. Bu ilişki GeoRSS ve RSS arasındaki ilişkiye oldukça benzerdir. Biçimlendirilmiş etiket yapısı kullanan KML ve GML gibi XML tabanlı dillerden farklı olarak JSON/GeoJSON hem insanlar hem de makineler tarafından kolayca okunup çözümlenebilmektedir. Bir GeoJSON nesnesinin sözdizimi oldukça basit isim/değer çifti yapısını izlemektedir (Şekil 3.6).

```
{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [125.6, 10.1]
  },
  "properties": {
    "name": "Dinagat Islands"
  }
}
```

Şekil 3.6 GeoJSON kodlamasında noktaların ifade şekli

Veri yapısını biçimlendirmede JavaScript dilinin bir alt kümesi olmasında rağmen GeoJSON dil-bağımsız bir yapıya sahiptir ve GeoJSON objelerinin sorgulanmasında ActionScript, Active Server Page, PHP vb. diğer teknolojilerle birlikte kullanılabilmektedir. Bir GeoJSON nesnesi bir geometriyi, bir elemanı (ek özelliklere sahip geometri) ya da eleman koleksiyonunu (eleman dizisi) temsil edebilmektedir (Howard Butler, 2016). GeoJSON birçok geometri türünü desteklemektedir. Bunlar arasında basit geometriler (nokta, çizgi, alan) ve geometrik kümeler (çoklu nokta, çoklu

[32] http://geojson.org/ adresinde detaylı bilgiye ulaşmak mümkündür.

doğru, çoklu çokgen, çoklu geometri gibi) bulunmaktadır. GeoRSS GML'de olduğu gibi standart WGS84 datumundan farklı koordinat referans sistemlerini de desteklemektedir. Topolojik özelliklerin kodlanması amacıyla GeoJSON formatından türetilen TopoJSON[33] formatı da mekânsal uygulamalarda kullanılan bir diğer formattır. Geometrileri ayrı parçalar olarak göstermek yerine TopoJSON dosyalarında geometriler "**arc**" adı verilen ortak doğru parçaları ile birbirine bağlanmaktadır (Github-contributors, 2016). **Arc**'lar nokta dizilerinden oluşurken çoklu doğrular ve kapalı alanlar **arc** dizilerinden oluşmaktadır (Şekil 3.7).

```
{
  "type": "Topology",
  "transform": {
    "scale": [0.036003600360036005, 0.017361589674592462],
    "translate": [-180, -89.99892578124998]
  },
  "objects": {
    "aruba": {
      "type": "Polygon",
      "arcs": [[0]],
      "id": 533
    }
  },
  "arcs": [
    [[3058, 5901], [0, -2], [-2, 1], [-1, 3], [-2, 3], [0, 3], [1,
1], [1, -3], [2, -5], [1, -1]]
  ]
}
```

Şekil 3.7 TopoJSON kodlaması örneği

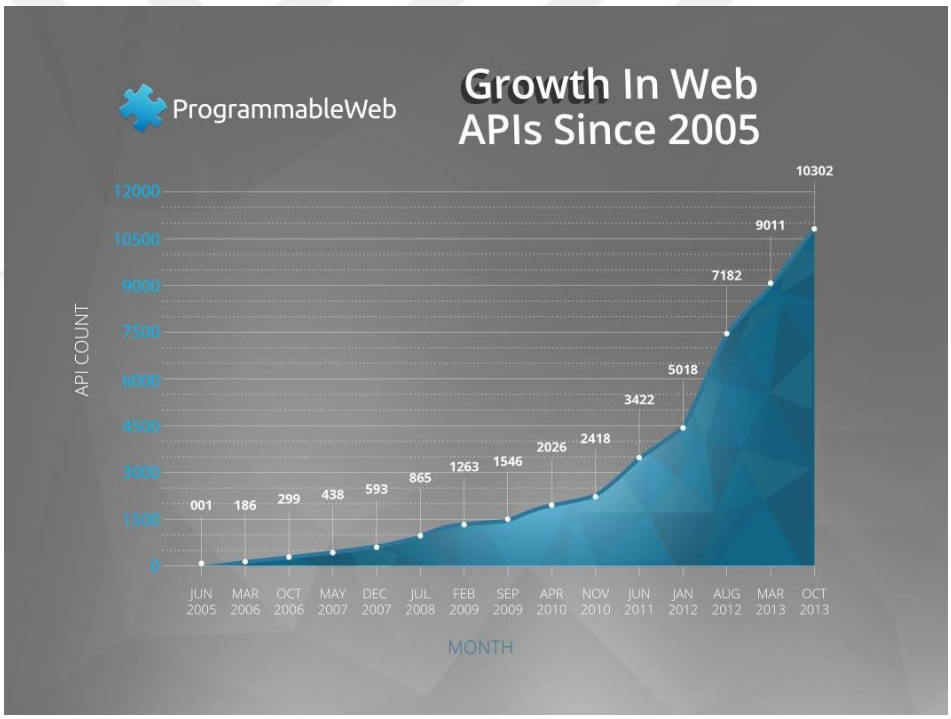
Her bir **arc** sadece bir kez tanımlanmakta ancak farklı şekiller tarafından birçok kez kullanılabilmektedir. Bu işlem tekrar eden nesnelerin önüne geçmekte ve dosya boyutunu oldukça düşürmektedir. Ek olarak TopoJSON, topoloji koruyan şekil basitleştirme, otomatik harita renklendirme ve kartogramlar gibi, topoloji kullanan uygulamaları hızlandırmaktadır.

3.3 API Kavramı

API kavramı Application Programming Interface ifadesinin baş harflerini temsil etmektedir. Bilgisayar programcılığında bir API, yazılım ve uygulamaların

[33] Detaylı bilgi, tanımlamalar ve örnek kodlamalar için https://github.com/mbostock/topojson/wiki/Introduction adresine bakılabilir.

oluşturulmasında kullanılan bir dizi rutin, protokol ve aracı temsil etmektedir. Web-tabanlı sistemler, işletim sistemi tabanlı sistemler ya da veri tabanı sistemleri için birçok API mevcuttur. Örneğin bir kullanıcı Android tabanlı bir uygulama oluşturacaksa, donanım ile yazılım arasındaki iletişim kurabilmek için bir Android API kullanmak zorundadır. Web geliştiricilerine mevcut programlama kütüphanelerine erişim ve oluşturacakları mashup uygulamalarına yeni işlevler ekleme imkânı sunmaktadır. API ara yüzleri, teknoloji firmalarının ürünlerini birbiri ile entegre etmek için kullandıkları en yaygın yoldur (Ternay, 2015). Web API ara yüzlerinin sayısı, ilk ortaya çıktıkları andan günümüze, hızlı bir artışla gelişimlerini sürdürmektedir. Ekim 2013 tarihinde bu rakam 10.302 adete ulaşmıştır (Şekil 3.8). Bu çalışmada harita mashup uygulamalarında yaygın olarak kullanılan haritalama API ara yüzleri ve diğer API ara yüzleri değerlendirilmeye çalışılmıştır.



Şekil 3.8 2005-2013 yılları arasında web API sayısındaki artış (www.programmableweb.com)

Web geliştirme alanında kullanıldığında bir API, genellikle XML ya da JSON (JavaScript Object Notation) içerisinde bir dizi Hypertext Transfer Protocol (HTTP) istek mesajı ve yanıt mesajı yapısı ile tanımlanmaktadır. Web API geçmişte web servisleri ile eş anlamlı kullanılmasına rağmen günümüz teknolojisi ile (web 2.0) Simple Object Access Protocol (SOAP) tabanlı web servisleri ve Service-Oriented Architecture (SOA) yapısından direkt REST tarzı web kaynakları ve Resource-Oriented

Architecture (ROA) yapısına yönelmiştir (Benslimane ve ark., 2008). Web API ara yüzleri, mashup adı verilen, çoklu API'leri yeni uygulamalar içerisinde birleştiren uygulamalara imkân sağlamıştır.

Web haritaları alanında API tarihçesi Google tarafından 2005 yılında duyurulan Google Maps ve hemen ardından dağıtımına başlanan, birçok haritalama fonksiyonu içeren, Google Maps API servislerine dayanmaktadır. Bu fonksiyonlar haritanın görünümünü, ölçek, konum ile nokta, çizgi ve alan biçiminde ek bilgilerin kontrolünü içermektedir. Ortaya çıkmasındaki temel amaç, kullanıcılara kendi web sayfalarına Google haritalarını ekleyebilme imkânı vererek, diğer web içerikleri ile haritaları birleştirme kabiliyeti sunmaktır. Ardından Yahoo!, Microsoft ve MapQuest gibi harita servis sağlayıcıları, çevrimiçi haritalama servislerini AJAX tabanlı kullanıcı ara yüzü içerecek şekilde değiştirmiştir. 2006 yılında Yahoo! Kendi API ara yüzünü yayınlamış, daha sonra da Microsoft haritalama servisini 2009 yılında şirketin arama motoru servisine aktarmış ve adını Bing Maps olarak değiştirmiştir. Diğer haritalama servislerine örnek olarak OSM ve ESRI verilebilmektedir. Her bir servisin kendine ait API ara yüzü bulunmaktadır ve bu sayede kullanıcılar farklı fonksiyonelliklere sahip bu servislerden dilediğini sayfalarına ekleyerek diğer servislerle ilişkilendirebilmektedir.

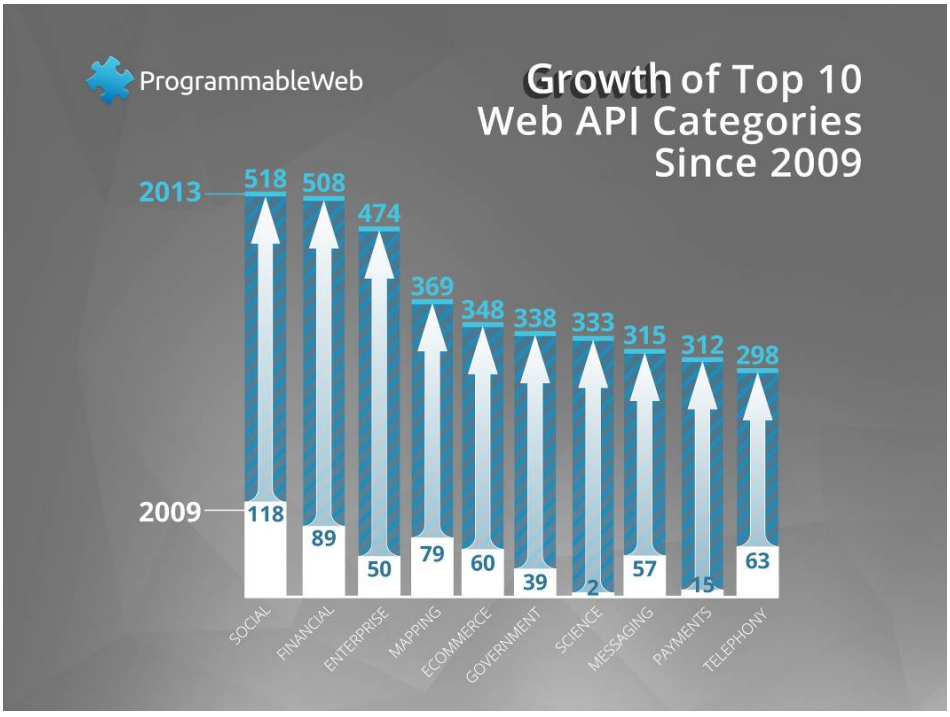
API'lerin gelişimi günümüzde halen başlangıç çağında sayılabilir ve daha birçok geliştirmeyle gelecekte karşılaşılması mümkündür. Ana servis sağlayıcılarında fonksiyonların çağırılması küçük farklılıklarla birbirine oldukça benzemesine rağmen her biri için kodların yeniden yazılması çok zaman almaktadır. Açık kaynaklı Mapstraction[34] API her bir haritalama API ara yüzü arasında geçiş yapılabilmesine imkân saplayan bir ara yüzdür ancak sadece temel fonksiyonları uygulayabilmektedir (DuVander, 2010). Google, haritalama API'lerinin geliştirilmesinde, haritalama servisine her geçen gün birçok yeni fonksiyon eklemeye devam ederek, lider konumunu korumaktadır (Peterson, 2012b).

3.3.1 Haritalama amaçlı API'ler

Haritalama API ara yüzleri, internet kullanıcılarının coğrafi veri ve mekânsal işlemler için güçlü harita sunucularına ulaşmasını sağlayan kaynak kodu ara yüzleridir. Coğrafi veriler hava ve uydu fotoğraflarının yanında nokta, çizgi alan gibi diğer özellikleri de içerebilmektedir. Mekânsal veri genellikle detaylı bir şekilde küresel çapta

[34] Ayrıntılı bilgi için bkz. http://mapstraction.com/

elde edilebilmektedir, ancak genellikle metropolitan alanlarda, kırsal kesimlere nazaran daha detaylı ve yüksek çözünürlüklü mekânsal veriye ulaşmak mümkündür. Sunucudan istenen API verisi genellikle istemciye WMS sistemlerinde olduğu gibi raster mozaikler olarak dönmektedir. Bunun yanında kullanıcılar uygun veri formatında özel verilerini bu raster veri üzerine ekleyebilmektedir. Bir haritalama API ara yüzü sayesinde ilgilenilen bir bölgenin verisi sunucudan istenebilir ve elde edilen sonuç başka bir web sitesi içerisinde görüntülenebilir. Bu sayede kullanıcılar konuma dayalı birçok uygulama geliştirebilmekte, kendi ihtiyaçları doğrultusunda haritalarını özelleştirebilmektedir. Bu olanaklar, son on yılda web haritalarına olan rağbet ve web haritalama araçlarının kullanımında dikkate değer bir artışa neden olmaktadır. Bu artışta mobil cihazlardan her an her yerde internete erişebilen kullanıcıların konum tabanlı uygulamaları günlük hayatlarında sıklıkla kullanmalarının payı büyüktür. Böylesine artış gösteren bir alanda gelişmiş uygulamaların görülmesi de kaçınılmazdır. Bunun sonucunda birçok yeni web-tabanlı haritalama servisi ortaya çıkmış ve API ara yüzlerini geliştirerek kullanıcılara sunmuştur. Web ortamında 2009 yılında 79 adet haritalama API ara yüzüne ulaşmak mümkünken 2013 yılının sonunda bu rakam 369'a çıkmış ve haritalama API'lerini en hızlı büyüyen API kategorisine yerleştirmiştir (Şekil 3.9).



Şekil 3.9 2009-2013 yılları arasında en hızlı büyüyen 10 web API kategorisi (programmableweb.com)

Her ne kadar web haritaları alanında Google lider konumda bulunsa da, günümüzde harita teknolojisi endüstrisinde söz sahibi diğer birçok şirket bulunmaktadır. Bu bölümde web üzerinde en çok kullanılan haritalama API ara yüzleri, yaygın kullanım, potansiyel, dokümantasyon, kullanım kolaylığı ve işlevsellik açısından değerlendirilmeye çalışılmıştır. Değerlendirmede kullanılan istatistiki bilgiler için *programmableweb.com* internet sitesinden ve Wagner (2015) tarafından gerçekleştirilen çalışmadan faydalanılmıştır.

3.3.1.1 Google Maps API

Ortaya çıktığında web haritalama alanında yeni bir çağ başlatan Google Maps API aradan geçen 11 yıla rağmen web haritaları alanında hâkimiyetini sürdürmektedir. Nielsen araştırma şirketine göre (2014) sadece Google Maps akıllı telefon uygulaması aylık 79 milyon bağımsız kullanım sayısına ulaşmaktadır. Bunun yanında Google tarafından oluşturulan resmi masaüstü Google Maps uygulaması bulunmakta ve milyonlarca üçüncü parti uygulama da Google haritalarını kullanmaktadır. Google Maps API üçüncü sürümünü (V3) yayınladığı genel haritalama API ara yüzünün yanında birçok farklı haritalama API ara yüzü de sunmaktadır. Bunlardan bazıları Embed API[35], Street View Image API[36], Android API[37], IOS API[38], Google Static Maps API[39], Directions API[40], Distance Matrix API[41], Elevation API[42] ve Geocode API[43] hizmetleridir. Google daha birçok API hizmeti sunmakta ve kullanıcılara kendi projelerine uygun haritalama API seçimine yardımcı bir liste de sağlamaktadır (GoogleDevelopers, 2016a). Bu hizmetler için oldukça detaylı bir API dokümantasyonu sunulmaktadır. Bu dokümantasyonlar kod örnekleri, kitaplıklar, SDK'lar[44] (Software Development Kit) ve diğer dijital haritalama araçlarını içermektedir. Google Maps API V3 KML, GeoJSON, GeoRSS ve Fusion Tables formatlarını direkt olarak desteklemektedir. Bu formattaki veriler harita üzerine katman olarak eklenebilmektedir.

[35] https://developers.google.com/maps/documentation/embed/
[36] https://developers.google.com/maps/documentation/streetview/
[37] https://developers.google.com/maps/documentation/android-api/
[38] https://developers.google.com/maps/documentation/ios-sdk/
[39] https://developers.google.com/maps/documentation/static-maps/
[40] https://developers.google.com/maps/documentation/directions/
[41] https://developers.google.com/maps/documentation/distance-matrix/
[42] https://developers.google.com/maps/documentation/elevation/intro
[43] https://developers.google.com/maps/documentation/geocoding/start
[44] SDK genel olarak belirli bir yazılım paketi için uygulama geliştirmeye izin veren bir dizi yazılım geliştirme aracıdır.

Google Maps JavaScript API, Google Static Maps API, Google StreetView Image API ara yüzlerini ardışık 90 gün süresince her 24 saatte 25.000 harita yüklemesi sınırı ile kullanıcılarına ücretsiz sunulmaktadır. Ayrıca diğer web servislerinin ücretsiz kullanımı için günlük 2.500 sorgulama sınırı getirmiştir. Bu sınırlamalar birçok ticari amaçlı olmayan uygulama için yeterli bir sayıdır.

3.3.1.2 BingMaps Control

Microsoft Bing Maps yaygın kullanılan bir haritalama platformudur. Ancak Google ile karşılaştırıldığında bu kullanım oldukça sınırlı kalmaktadır. Microsoft şirketi Bing Maps platformuna yeni özellikler ve geliştirmeler eklemekte ve eklemeye devam etmektedir. Microsoft Bing Maps API, kullanıcılarına oldukça geniş bir dokümantasyon sunmaktadır. Ancak bu dokümanları ve örnek çalışmaları takip etmek oldukça zordur. Ancak bunun yanında, web geliştiricileri için JavaScript uygulamaları oluşturmada, etkileşimli AJAX tabanlı SDK ile sunulan özellikler ve kod örnekleri de mevcuttur.

Bing Maps, Microsoft MapPoint ve Terra Server gibi mevcut Microsoft teknolojileri üzerine kurulmuştur. Orijinal sürümü, günümüzde ayırt edici özellikleri olan bird's eye view ve 3 boyutlu haritalardan yoksundu ve sadece ilgi alanı noktalarını gösteren basit bir katman içeriyordu. 2005 Kasım ayında ortaya çıktığından bu yana Windows Live Local, Virtual Earth platformunun halka sunulan yüzü olmuştur. 2006 yılında Microsoft haritaları 3 boyutlu olarak görüntüleyebilen yeni bir özellik ekleyerek bu ürünün ismini Live Search servislerini ekleyerek "Live Search Maps" olarak değiştirmiştir. 2009 yılından günümüze kadar da bu platform Bing Maps olarak devam etmektedir. Bing Maps platformu, kullanıcıların uygulamalarında kullanabileceği, AJAX control, Windows Store Apps control, Windows Presentation Foundation control, REST servisi ve Mekânsal Veri Servisi şeklinde isimlendirdiği birçok API seçeneği sunmaktadır. Özellikle AJAX v7 API ara yüzü web tabanlı uygulamalar için geliştirilmiştir.

3.3.1.3 MapQuest OpenAPI

MapQuest sayısal harita üretimi için MapQuest.com web sitesi, MapQuest Mobile, MapQuest Local, MapQuest Enterprise ve MapQuest Developers gibi oldukça zengin seçenekler sunmaktadır. MapQuest platformunun iki farklı sürümü mevcuttur.

Bunlardan ilki lisanslı ikincisi de açık kaynaklıdır. Lisanslı veri platformu bir MapQuest Enterprise Edition lisansı ile alınabilmekte, açık kaynaklı veri ise ücretsiz ve açık lisansla erişilebilmektedir. MapQuest Open Data platformunun ana veri kaynağı OpenStreetMap tarafından sağlanmaktadır. MapQuest geliştiriciler için oluşturduğu web sitesinde kolaylıkla ulaşılabilen dokümantasyonlar sunmaktadır. Bunun yanında bir harita oluşturma aracı, bir statik harita sihirbazı, bir güzergâh planlayıcı ve diğer geliştirici araçlarını sunmaktadır. Ek olarak MapQuest bir de JavaScript Maps API ara yüzü sağlamakta, bu ara yüzle Open Web Service, Open Leaflet plug-ins ve daha fazla araca ulaşmak da mümkün olmaktadır.

Aslında MapQuest, henüz Google şirketi kurulmadan önce 1967 yılında oluşturulmuştur. Google, 2005 yılında Google Maps hizmetini başlatmadan önce MapQuest bir süre sayısal haritalama araçları sunmuştur. MapQuest platformu iyi tasarlanmış, güvenilir ve gücünü milyonlarca MapQuest.com ziyaretçisi ve MapQuest app kullanıcısından almaktadır (Wagner, 2015).

3.3.1.4 ArcGIS JavaScript API

ESRI 1969 yılında arazi kullanım planlamaları için coğrafi verileri analiz eden bir araştırma grubu olarak oluşturulmuştur. 1996 yılında Internet üzerinde harita yayınlamak amacıyla ESRI, MapObjects adındaki ilk platformunu yayınlamıştır. Günümüzde ise ESRI temel olarak kuruluşlara odaklanmakta ve bütüncül bir CBS platformu sunmaktadır. Bu platform ArcGIS bulut tabanlı haritalama platformu, masaüstü ArcGIS Desktop yazılımı, ESRI City Engine ve ESRI MapStudio ürünlerini kapsamaktadır. ESRI tamamıyla ticari CBS pazarına yönelse de ArcGIS Explorer Desktop adında ücretsiz bir CBS görüntüleme aracı da geliştirmiştir.

ESRI geliştiriciler için oldukça gelişmiş ve takip etmesi kolay bir portal oluşturmuştur. Geliştirici portalı ArcGIS API ara yüzlerine SDK'lara, organize dokümantasyona, kod örneklerine ve daha birçok yardım dokümanına erişimi kapsamaktadır. ESRI geliştiriciler için kredi sistemi kullanan bir ücretlendirme sunmaktadır.

3.3.1.5 OpenLayers JavaScript API

OpenLayers WebGL, Canvas 2D ve diğer HTML5 özelliklerini modern web tarayıcılarında görüntüleyebilmek için kullanan açık kaynaklı bir JavaScript kütüphanesidir. OpenLayers OSM, Bing Maps, MapQuest ve diğer birçok harita kaynağı üzerinden mozaik görüntü çekebilme yeteneğine sahiptir. Bunun yanında GeoJSON, TopoJSON, KML, GML, GML ve diğer coğrafi veri formatlarından vektör veri işleme imkânı sunmaktadır. OpenLayer için Internet üzerinden ulaşılabilen iyi organize edilmiş dokümantasyon sunulmakta ve geliştiriciler için çok sayıda bilgi sağlanmaktadır. OpenLayer web sitesinde bulunan galeride çok seçenekli örnek uygulamalar ve kod örnekleri mevcuttur.

OpenLayer API açık kaynak lisansı ve diğer harita kaynaklarından mozaik görüntü çekebilme kabiliyeti ile yaygın olarak kullanılan bir haritalama kütüphanesidir.

3.3.1.6 Carto DB

CartoDB, geliştiricilere web ve mobil mekânsal uygulamalar geliştirebilmek için bir açık kaynak haritalama, analiz ve görselleştirme motorudur. Nisan 2012 tarihinde tanıtılmıştır ve o günden bu yana National Geographic, NASA, The Guardian UK ve Twitter gibi yüksek profilli kullanıcılar kazanmıştır. CartoDB ücretsiz temel sürümünden kurumsal çözümler için lisanslı sürüme bir çok seçenek sunmaktadır.

CartoDB, geliştiricilerin kendi uygulamalarına etkileyici haritalar ve mekânsal görselleştirmeler ekleyebilecekleri bir JavaScript kütüphanesi (CartoDB.js) ve birçok API sağlamaktadır. Bunun yanında kullanıcıların hızlı bir şekilde harita oluşturmalarına ve veri görselleştirmelerine imkân veren sezgisel bir sürükle-bırak ara yüze sahip editör sunmaktadır. CartoDB ve editör için iyi organize edilmiş dokümantasyonlara ulaşmak ve takip etmek kolaydır.

Oldukça yeni bir platform olmasına rağmen CartoDB çok hızlı bir büyüme yaşamıştır. En popüler özelliklerinden biri mekânsal verinin zaman göre değişimini animasyon şeklinde görselleştirebilen Torque platformudur. Bu gibi özellikleri CartoDB platformunu diğer sayısal haritalama ve mekânsal platformlardan ayırmaktadır.

3.3.2 Diğer API'ler

Dinamik olarak çoklu veri kaynaklarını bir araya getirerek oluşturulan mashup uygulamaları çoğunlukla haritalama amaçlı olmayan ve konum tabanlı içerik sunan API ara yüzlerini de kullanmaktadır. Mashup uygulamaları daha önce de belirtildiği üzere tamamen geliştiricisinin hayal gücü ile ilişkilidir. Çünkü gerek haritalama API ara yüzleri, gerekse diğer ara yüzler sınırsız sayıdadır. Mashup uygulamalarına eklenmek istenen veriler çoğu zaman direkt olarak bir haritalama motoruna entegre edilebilecek uygun formatta (örn., GML, KML, GeoRSS) konum bilgisine sahip olmamaktadır. Örneğin Google Fusion Tables API, kullanıcıların özel verilerini (.csv, .txt, KML, .xls gibi) Fusion tablolarına dönüştürebilmekte ve bu veriyi konum bilgisi içeriyorsa harita üzerinde gösterebilmektedir. Fusion Tables Google'ın bir servisi olduğu için Google haritalarını kullanmaktadır ve coğrafi referansı WGS84 datumudur. Eğer tabloda her bir nesneye ait koordinat bilgisi bu referans sistemine uygun değilse Google geocode servisi kullanılarak adres, şehir adı, mevki gibi bilgilerden koordinat üretilebilmektedir. Ancak bu, kullanılan servisin kapsamına bağlıdır ve her zaman istenen sonuçlar alınamayabilir. Buna ek olarak Internet üzerinde en az bir konum bilgisi içeren veriler için koordinat üreten ücretsiz servisler de mevcuttur. Haritalama amaçlı olmayan API'ler harita mashup uygulamalarıyla kullanıldıklarında ortaya çıkan en büyük problem mekânsal referansa sahip olmayan veridir.

Aslında Chow (2011) tarafından: *"web CBS uygulamaları için coğrafi veri (örn., harita) ve mekânsal işlemleri (örn., yön tarifi) içeren çoklu üçüncü-parti coğrafi bilgi servislerinin entegrasyonunu sağlayan mashup uygulamalarının dağıtımı"* olarak tanımlanan ve kökenini O'Reilly 2006 konferansından[45] alan Coğrafya 2.0 kavramı kapsamında, haritalama API ara yüzleri hakkında gerçekleştirilen çalışmalar oldukça sınırlıdır. Peterson (2015) haritalama API ara yüzlerinin bir değerlendirmesini yapmıştır. Ona göre haritalama API'leri değerlendirilirken farklı birçok kriter belirlenebilmektedir. Ancak kullanım maliyetleri ve servisin ömrünün en temel değerlendirme kriteri olması gerektiğinin altını çizmektedir. Bunun en önemli örneklerinden biri Yahoo! Maps'dir. Bu tezin yayınlandığı tarihlerde Yahoo! kısa bir süre sonra haritalama faaliyetlerine son vereceğini ve tüm harita işlemlerini ortağı olduğu Nokia şirketine devredeceğini açıklamıştır. Yahoo! Maps API ile oluşturulan

[45]' O'Reilly Emerging Technology Conference' başlıklı konferans arşivine http://www.oreilly.com/conferences/ adresinden ulaşılabilmektedir.

uygulamaların yeni bir platform için yeniden uyarlanması gerekmektedir ve bu zor ve zahmetli bir iştir. Ayrıca taşınacak yeni platformun aynı işlevsellikleri sunabileceği de kesin değildir. Mielke ve Burger (2007) tematik kartografya bakış açısıyla haritalama API ara yüzlerinin kısıtlamalarının altını çizmektedir. Kullanıcılar genellikle web-tabanlı haritalama servisleri kullanarak oluşturulan mashup uygulamalarında, servisin sunduğu tasarım öğeleri ile yetinmek zorunda kalmaktadır. Farklı servislerin bu konuda farklı çözümleri mevcuttur. Örneğin Google Adobe Photoshop ile oluşturulan işaretlerin harita üzerine eklenmesine imkân vermektedir ayrıca tematik haritalar konusunda daha zengin seçenekler sunan CartoDB, çok daha fazla özelleştirme imkanı sunabilmektedir. Masaüstü uygulamalar için de aynı durum geçerlidir. Tek farkı çok daha zengin özel işaret kütüphanelerine sahip olmalarıdır ya da yazılıma entegre özel işaret oluşturma araçları sunabilmektedir.

API değerlendirme kriterleri belirlenirken aslında kullanım amacı da ön plana çıkmaktadır. Örneğin bu çalışmada dış veri ile entegrasyon en önemli konulardan biridir. İncelenen API ara yüzleri içerisinde Google, ESRI, CartoDB haricindekiler üstün harita yükleme ve farklı amaçlar için farklı seçenekler sunsa da (örn., yol tarifi) dış veri ile entegrasyonda yetersiz kalmaktadır. Bu çalışmada kullanılacak API seçilirken öncelikle dış veri ile entegrasyon, harita kapsamı, yükleme hızları, servisin yayında olduğu süre ve kullanıcı sayısı (servisin olgunluğu[46]) kriterleri dikkate alınmıştır. Google, 10 yılı aşkın haritalama geçmişine sahiptir ve bu akımın öncüsüdür. Dünya'da kullanıcı sayısının en fazla olduğu haritalama servisine sahip olduğu birçok kaynakta resmiyet kazanmıştır (Richter, 2013). Mekânsal veri altyapısı güncel, sürekli güncellenmeye devam etmekte ve tüm dünyayı kapsamaktadır. Dış veri ile entegrasyon konusunda, elinde bulundurduğu bir çok servisi de kullanarak farklı çözümler sunmaktadır. Örneğin Google bünyesinde bulunan Fusion Tables API özellikle kullanıcıların özel verilerini tablo yapısında korumakta ve gerektiğinde Google haritaları üzerine katman olarak ekleme imkânı sunmaktadır. Ayrıca W3C tarafından 2008 yılında bir standart olarak kabul edilen KML formatı Google tarafından ortaya atılan bir formattır ve en gelişmiş KML desteği Google haritalar tarafından sunulmaktadır. Bunun yanında GeoJson, GeoRSS gibi formatları da desteklemektedir. Bireysel kullanıcılar için tamamen ücretsizdir, getirilen kısıtlamalar, oluşturulan

[46] Bir servisin yayında olma süresi ve kullanıcı sayısı, servis için oluşturulan dokümantasyon arşivi, servisin talepler üzerine getirdiği güncellemeler ve ek fonksiyonlar ve geleceğe yönelik servisin ömrü konularında etkileyici unsur olarak düşünülmüştür. Bu kritere, bu çalışmada "servisin olgunluğu" olarak atıf yapılmıştır.

uygulamalar ticari amaçlı olmadıkça ulaşılması zor sınırlamalardır. Ayrıca yukarıda belirtildiği üzere sayısız servisinin gücünü de haritaları ile birleştirebilmektedir. Örneğin Google Drive servisi sayesinde kullanıcılar oluşturduğu uygulamalarını bu serviste tutabilmekte ve sunucu olarak kullanabilmektedir. Uygulamalarda Google haritalarının kullanılmasının bir düğer sebebi de Google haritalarının sezgisel kullanıma uygun olması ver firmanın gerçekleştirdiği diğer çalışmalar ya da anlaşmalar sebebiyle (örn., Google tarafından geliştirilen Android işletim sistemi sayesinde Dünya üzerinde milyonlarca kullanıcı mobil cihazlarında konuma dayalı işlerini Google haritalar sayesinde yürütmektedir) potansiyel kullanıcıların Google haritalarına aşina oluşudur. Bu da kullanım kolaylığı açısından bir avantaj olarak değerlendirilebilmektedir. Kullanıcı sayısının gün geçtikçe artması ve ticari politikaları da dikkate alınarak Google firmasının haritalama sektöründe daha uzun süre hakim olarak devam edeceği düşünülmektedir. Bu da servisin yakın bir tarihte kapatılmayacağı açısından bir öngörüdür.

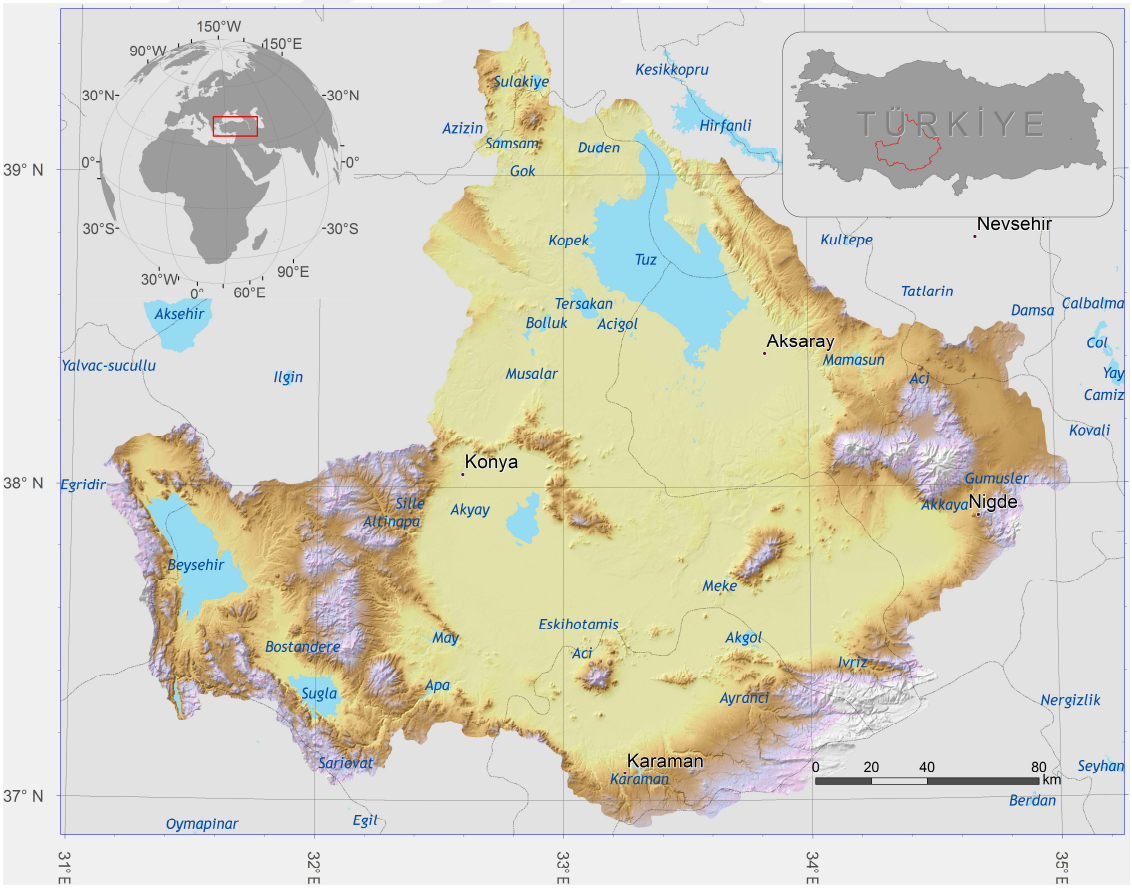
Yukarıda açıklanan nedenlerden dolayı bu çalışmada Google Maps ve Google tarafından oluşturulan diğer servisler tercih edilmiştir. 5. bölümde bu servislerin kullanılmasında karşılaşılan problemler ve avantajlar detaylı olarak tartışılmıştır.

4 DURUM ÇALIŞMASI: KKH MASHUP UYGULAMASI

Bu bölümde, Konya Kapalı Havzası mashup uygulaması durum çalışması ele alınmıştır. İlk olarak çalışma bölgesi tanıtılmıştır. Ardından bu bölgede zemin çökmeleri ve nedenlerinin araştırılması için kullanılacak veri seti detaylandırılmıştır. Oluşturulan harita mashup uygulaması; 1) grafik ara yüz tasarımları, 2) veri katmanları, 3) dinamik lejant yaklaşımı, 4) zaman kaydırma çubuğu yaklaşımı, 5) istatistiksel yüzey (izoplet harita katmanı) yaklaşımı ve 6) katman karşılaştırma "Swipe Mode" ara yüzü, başlıkları altında yine bu bölümde ele alınmıştır.

4.1 Çalışma Bölgesi

Çalışma bölgesi olan KKH, Türkiye'nin İç Anadolu Bölgesi'nde yer almaktadır (Şekil 4.1). Türkiye'nin toplam alanının %7'sine denk gelen 50,000 km^2'lik bir alanı kaplamaktadır.

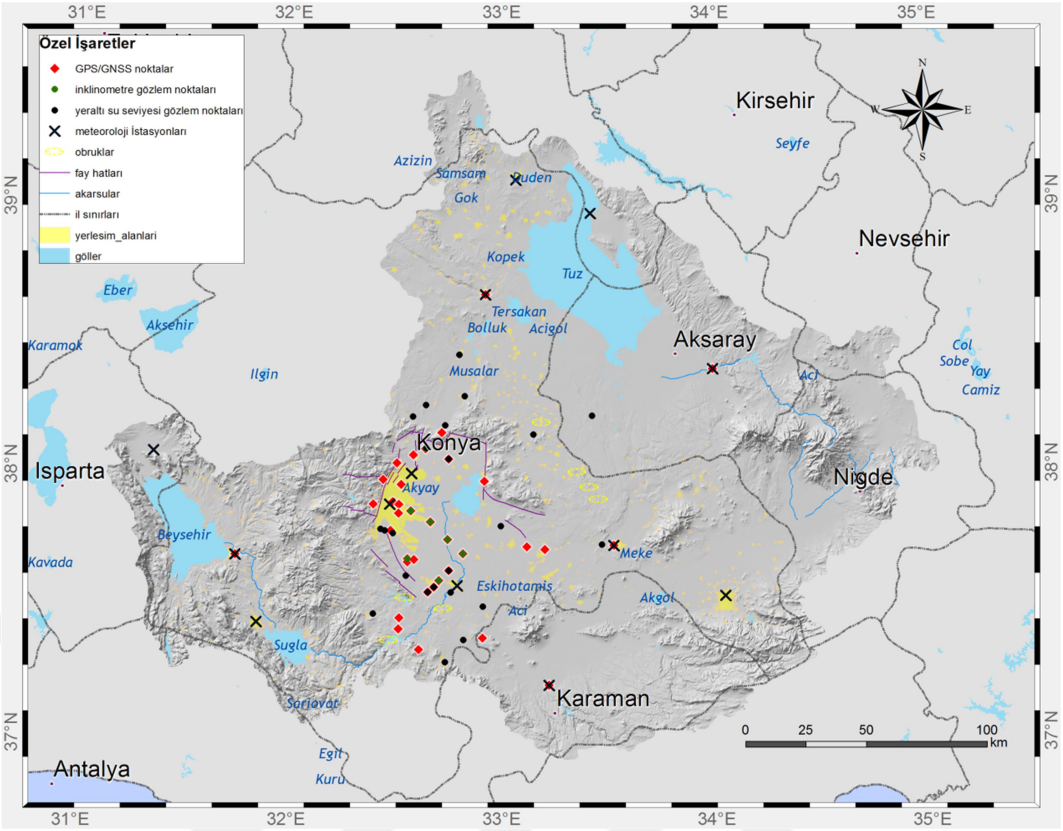


Şekil 4.1 Konya Kapalı Havzası çalışma bölgesi

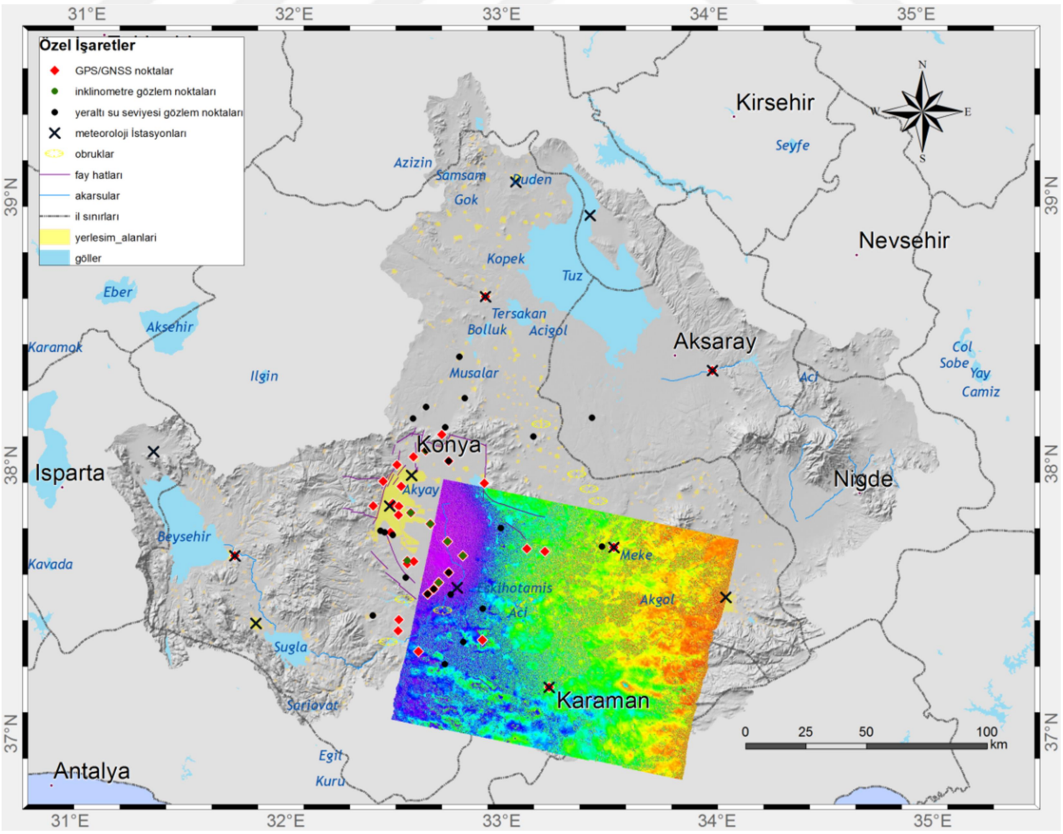
Kapalı havzalar, sularını denizlere kadar ulaştıramayıp kuruyan, yer altına sızdıran veya göle dökülüp kalan akarsuların oluşturduğu bölgelerdir. Kapalı havzaların oluşmasında; yer şekillerinin oluşumu ve iklim etkilidir. Kapalı havzalar genellikle iç kesimlerde, kurak iklim bölgelerinde görülür. Açık havza, sularını akarsular aracığıyla deniz ve okyanus gibi açık sulara ulaştıran havzalardır. Önünde yüksek dağlar olmayan, aşırı kurak olmayan ve karstik olmayan alanlar genellikle açık sulara ulaşabilirler. Deniz kıyısına yakın yerler ve nemli alanlar da genellikle açık havzalıdır.

4.2 Kullanılan Veri Seti

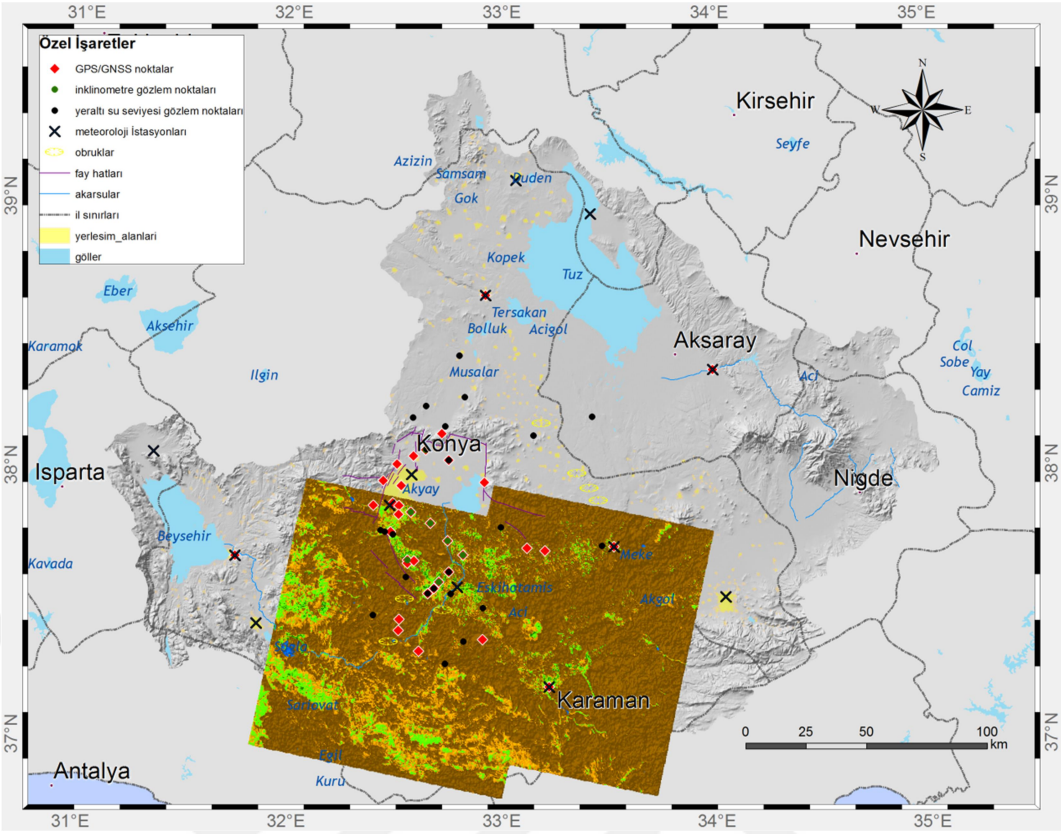
Konya Ovası içerisinde yer kabuğu hareketlerinin izlenmesi ve nedenlerinin araştırılması kapsamında birçok farklı kaynaktan veri toplanmıştır. İlk veri seti, GNSS uyduları yardımıyla 3 boyutlu yeryüzü hareketlerinin izlenmesi amacıyla, sabit noktalar da dâhil olmak üzere (TUSAGA-Aktif) 28 noktadan oluşan ağda gerçekleştirilen gözlemleri içermektedir. Bunun yanında ikinci veri seti olarak, yeraltı su seviyelerinin yer kabuğu hareketleriyle bağlantısının araştırılması için Devlet Su İşleri (DSİ) tarafından kullanılan 11 yer altı su seviyesi izleme noktası ve bu noktaların gözlem değerleri bulunmaktadır. Ayrıca aynı 11 noktada inklinometre ve manyetik oturma gözlemleri de gerçekleştirilmiş olup, kullanılan inklinometrik gözlem noktaları gözlem değerleri de bir diğer veri setini oluşturmaktadır (Ustun ve ark., 2010) (Şekil 4.2). Ayrıca, çalışma bölgesinde meydana gelen yatay ve düşey yönlü hareketlerin incelenmesinde GPS/GNSS gözlem verilerinin yanında InSAR görüntü çiftleri de kullanılmıştır. Haziran 2002 ve Haziran 2009 tarihlerini kapsayan ENVISAT uydusuna ait yaklaşık 50 radar görüntüsü, Avrupa Uzay Ajansı'ndan (ESA) temin edilmiş, görüntü çiftleri (Şekil 4.3) ve 2003-2010 yılları için oluşturulan normalleştirilmiş fark bitki endeksi (NDVI) tematik haritaları da (Şekil 4.4) veri setine dâhil edilmiştir. Ayrıca, zemin çökmesi ve yeraltı su seviyeleri arasındaki bağlantıların modellerinin araştırılması için, ek olarak, bölgede bulunan meteorolojik istasyonların Türkiye Meteoroloji Genel Müdürlüğü'nden (MGM) temin edilen yağış verileri de veri setine eklenmiştir. Daha önce de belirtildiği üzere, bölgede tarımsal sulama amaçlı açılan sondaj kuyularının sayısı oldukça fazladır (~100.000 adet). Yeraltı su seviyesi değişimlerinde bu durumun önemli bir etkisi olduğu düşünülmüş ve analizlerde kullanılabilmesi amacıyla DSİ bölge müdürlüğünden kuyu konum verileri alınarak veri setine dâhil edilmiştir (Şekil 4.5).



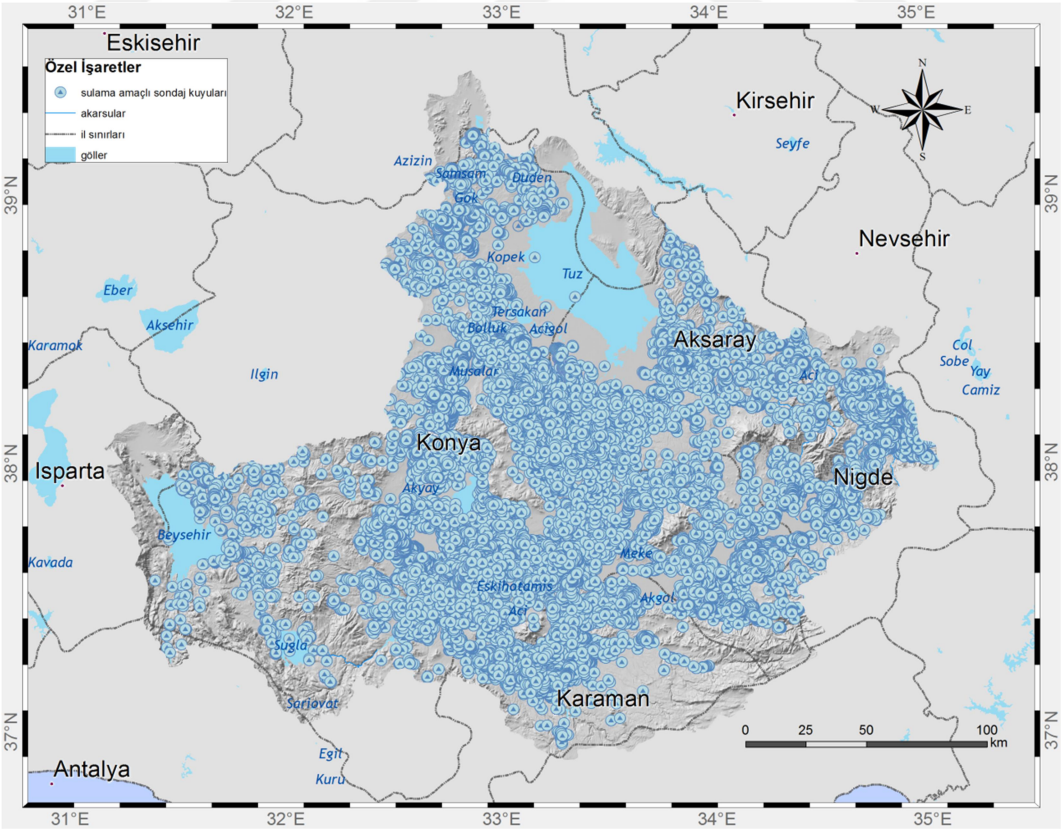
Şekil 4.2 Çalışma bölgesinde deformasyon izleme çalışmalarında kullanılan gözlem istasyonları dağılımı



Şekil 4.3 Çalışma bölgesi için kullanılan InSAR enterferogramları



Şekil 4.4 Çalışma bölgesi için kullanılan normalleştirilmiş fark bitki endeksi (NDVI) görüntüleri



Şekil 4.5 Çalışma bölgesinde bulunan sulama amaçlı sondaj kuyuları

Çalışmada kullanılan tüm mekânsal nesneler ve içerikleri Tablo 4.1'de, bu verilere ait öznitelik bilgileri de Tablo 4.2'de özetlenmiştir. Farklı kaynaklardan toplanan verilerin depolanması, yönetimi ve analizi işlemleri, bir sonraki bölümde detaylandırılacaktır.

Tablo 4.1Çalışmada kullanılan mekânsal nesneler ve içerikleri

Obje	Geometri Tipi	İçerik
GPS	Nokta	Nokta konumları
Yeraltı su seviyesi ölçüm noktaları	Nokta	Nokta konumları
İnklinometrik ölçüm noktaları	Nokta	Nokta Konumları
Meteorolojik gözlem istasyonları	Nokta	Nokta konumları
Yerleşim alanları	Alan	Çalışma bölgesinde bulunan yerleşim alanları
Jeolojik veriler	Alan, Çizgi	Jeolojik katmanlar ve fay hatları
Hidrojeolojik veriler	Nokta, Alan, Çizgi	Hidrojeolojik katmanlar, yer altı su kuyuları
Konya Kapalı Havzası sınırı	Çizgi	Konya Kapalı Havzası sınır çizgisi
Yerleşim merkezleri	Nokta	Çalışma bölgesindeki yerleşim merkezi konumları
SRTM sayısal yükseklik modeli	Raster	Çalışma bölgesini kapsayan 90 m aralıklı 3 boyutlu nokta kümesi
Deformasyon enterferogramları (InSAR)	Raster	Çalışma bölgesini kapsayan ve yüzey değişimlerini gösteren nokta kümesi

Tablo 4.2 Çalışmada öznitelik bilgileri kullanılacak nesneler ve elde ediliş biçimleri

Obje	Öznitelik İçeriği	Elde ediliş biçimi
GPS ve GNSS	Noktalara ait yer değiştirme ve hız verisi	Periyotlar halinde ölçüm yapılarak
Yer altı su seviyesi ölçüm noktaları	Yer altı su seviyesi değerleri	Devlet Su İşleri Genel Müdürlüğü
İnklinometrik ölçüm noktaları	Zeminde meydana gelen manyetik hareket verisi	Sürekli gözlem yapılarak
Meteorolojik gözlem istasyonları	Çalışma bölgesinde bulunan meteorolojik istasyonlardan (sinoptik) elde edilen aylık toplam yağış verisi	Meteoroloji İşleri Genel Müdürlüğü

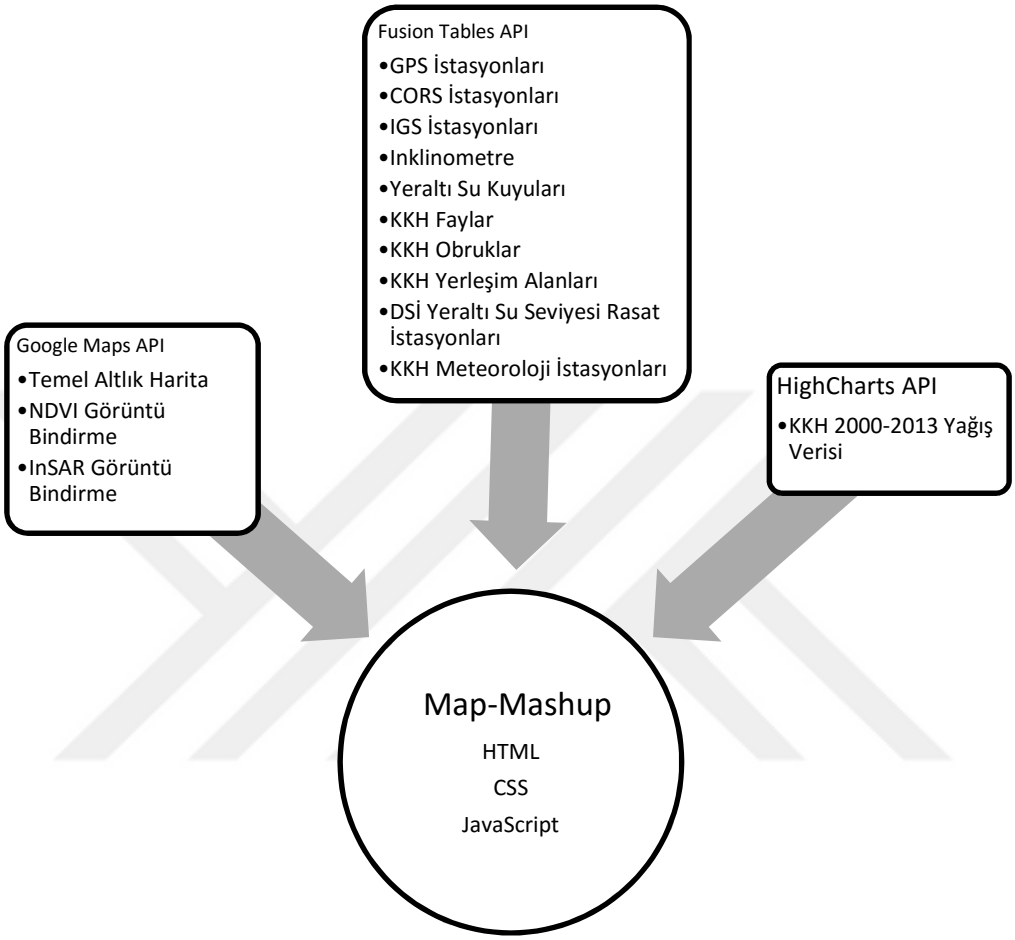
4.3 Harita Mashup Uygulaması

Bu bölümde oluşturulan mashup uygulaması, kullanılan altlık harita, grafik ara yüz tasarımı, veri katmanları, dinamik lejant yaklaşımı, zaman kaydırma çubuğu, istatistiksel yüzey yaklaşımı ve katman karşılaştırma ara yüzü başlıkları altında ele alınmıştır. Oluşturulan mashup uygulama altyapısı Şekil 4.6'da gösterilmektedir.

4.3.1 Altlık Harita: Google Maps

Oluşturulan mashup uygulaması için altlık olarak Google haritaları kullanılmıştır. Google, web harita servislerinde ve web üzerinde görüntü dosyalarının paylaşımında sıklıkla kullanılan mozaikleme yöntemini kullanmaktadır. Bir mozaik web haritası ya da mozaik görüntü, internet üzerinden çağırılan ve birbirlerine aralıksız olarak birleşebilen

çok sayıda bireysel görüntü parçasının, web tarayıcısında bir araya gelerek görüntülenmesi mantığı ile çalışmaktadır. Haritaların web üzerinde gösterimi ve harita üzerinde gezinmek için kullanılan en yaygın yöntemdir.



Şekil 4.6 Harita mashup uygulaması altyapısı

WMS gibi tek bir büyük harita görüntüsünün görüntüleyen ve üzerinde gezinmek için ok tuşları kullanan diğer yöntemlerden çok daha kullanışlıdır. Google Maps bu tekniği kullanan ilk büyük harita sitesinden biridir. Aynı yöntem vektör yapıdaki mozaikler için de kullanılabilmektedir ve bir süre sonra tamamen vektör mozaiklerin, imaj mozaiklerin yerini alması mümkün gözükmektedir (Roth ve ark., 2015).

Mozaik yöntemin birçok temel avantajı vardır. Kullanıcı harita görüntüsünün her kaydırdığında, görüntülerin büyük bir bölümü ekranda kalmaktadır ve görüntülenmeye devam etmektedir. Sadece görüntülenen alana yeni giren mozaik parçaları yüklenmektedir. Bu olay, haritanın tamamının tek bir görüntü olarak yüklenmesi ve

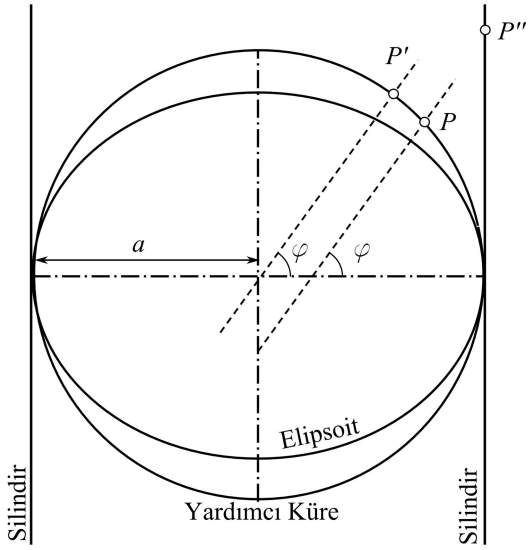
görüntülenmesi olayına kıyasla, kullanıcı deneyimini oldukça geliştirmektedir. WFS gibi teknolojilerde haritanın tamamının aynı anda yüklenmesi çok uzun zamanlar alabilmekte (haritanın kapsama alanı ve çözünürlüğü gibi parametrelere bağlı olarak) ve kullanıcı harita üzerinde gezinirken sıklıkla haritanın bütün olarak yeniden yüklenmesi gerekmektedir. Belirli kurallar ve standartlar gerektiren mozaik web haritalarının özellikleri mozaik boyutları, zoom seviyesi numaralandırması, kullanılacak projeksiyon, her bir mozaik parçasının numaralandırılmasında ya da tanımlanmasında izlenen yol ve bu parçaları çağırmak için kullanılan yöntemleri içermektedir.

Birçok mozaik web haritası belirli Google Maps kurallarını takip etmektedir. Bu kurallar aşağıdaki gibidir;

- Mozaik parçaları 256x256 pikselden oluşmaktadır,
- En yüksek zoom seviyesi 0 olup tüm dünya haritası tek bir görüntüde gösterilmektedir,
- Her bir zoom seviyesinde mozaik parçalarının boyutları iki katına çıkmaktadır. Bunun sonucunda her bir mozaik parçası, bir kez yakınlaştırıldığında yerine 4 adet mozaik parçası gelmektedir. 19 zoom seviyesine sahip bir web haritası bu nedenle birçok görev için oldukça yeterlidir,
- Haritalarda Web Mercator (geleneksel Mercator projeksiyonunun değiştirilmiş hali) projeksiyonu kullanılmaktadır, ancak projeksiyonun doğası gereği kutup bölgelerinin gösterimi mümkün olmamaktadır, $\sim\pm85^{o}$ arası görüntülenir.

Web-tabanlı haritalama servisleri Mercator Projeksiyonunun değiştirilmiş bir hali olan Web Mercator projeksiyonunu kullanmaktadır. Bu kullanımda WGS84 elipsoidi küresel projeksiyon eşitlikleri kullanılarak düzleme aktarılmaktadır. Yani bu durumda çift projeksiyon kullanımı söz konusudur (Şekil 4.7).

Sonuçta üretilen projeksiyonun artık konform (açı koruma) özelliği bozulmuş olmaktadır. Daha yalın bir ifadeyle Web Mercator, referans yüzey olarak elipsoit kullanan Mercator projeksiyonundan farklı olarak, referans yüzey olarak küre kullanmaktadır.



Şekil 4.7 Web Mercator projeksiyonunda çift projeksiyon kullanımı.

Referans yüzeyinden haritaya izdüşüm;

$$x = x(\varphi, \lambda) \tag{4.1}$$

$$y = y(\varphi, \lambda) \tag{4.2}$$

şeklinde ifade edilmektedir. Silindirik projeksiyonlarda eşitlikler basitleşerek;

$$x = x(\lambda) \tag{4.3}$$

$$y = y(\varphi) \tag{4.4}$$

şeklini almaktadır. Burada x ve y harita koordinatlarını (projeksiyon koordinatları) φ ve λ ise coğrafi koordinatları temsil etmektedir. Çünkü harita alanı dikdörtgen şekline dönüşmekte ve x değeri boylamın y değeri de enlemin birer fonksiyonu haline gelmektedir. Mercator projeksiyonunda, referans yüzeyi küre alındığında ve silindirin küreye teğet olduğu durumda (tek standart paralel) bir noktanın koordinatları;

$$x = R.\lambda \tag{4.5}$$

$$y = R \ln \tan\left(\frac{\pi}{4} + \frac{\varphi}{2}\right) \tag{4.6}$$

şeklinde hesaplanmaktadır. Burada R, referans kürenin yarıçapını temsil etmektedir. Referans yüzeyinin elipsoit alınmasında ise, bir noktanın koordinatları;

$$x = a\lambda \tag{4.7}$$

$$y = a \ln \tan\left(\frac{\pi}{4} + \frac{\varphi}{2}\right)\left(\frac{1 - e \sin \varphi}{1 + e \sin \varphi}\right)^{\frac{e}{2}} \tag{4.8}$$

şeklinde hesaplanmaktadır. Burada "e" elipsoidin 1. Dış merkezliği, a ise büyük yarı eksenidir . Web Mercator projeksiyonunda ise elipsoidal enlem ve boylam değerleri kullanılmasına rağmen koordinatlar referans yüzey küreymiş gibi hesaplanmaktadır. Bu durumda küre yarıçapı olarak WGS84 elipsoidinin büyük yarı ekseni kullanılmaktadır. Yani (4.5) ve (4.6) formülleri;

$$x = a\lambda \tag{4.9}$$

$$y = a \ln \tan\left(\frac{\pi}{4} + \frac{\varphi}{2}\right) \tag{4.10}$$

şekline dönüşmektedir. Aitchison (2011) bu durumu şu şekilde yorumlamaktadır:

"birçok çevrimiçi haritalama sistemi (Google Maps, Bing Maps, ArcGIS Online vb.) tarafından kullanılan Web Mercator projeksiyonu coğrafi koordinatları WGS84 datumu kullanarak tanımlamaktadır, ancak bu koordinatları bir küre üzerinde tanımlanmış gibi izdüşürmektedir".

Bu nedenlerden dolayı projeksiyon bazı uluslararası kaynaklarda "Spherical Projection" olarak isimlendirilmektedir. İki projeksiyon arasındaki fark haritaya gözle görülecek kadar büyük bir değişim getirmese de veri görüntüleme ve analizler konusunda kuzey ve güney uç noktalarda aradaki fark 50km'ye kadar çıkabilmektedir. Yapılan modifikasyon nedeniyle (küresel enlemin elipsoidal enleme eşit alınması nedeniyle) Web Mercator projeksiyonu konform özelliğe sahip değildir. Analizler ve daha detaylı karşılaştırma sonuçları için Battersby ve ark. (2014) tarafından gerçekleştirilen çalışmadan ve Bildirici (2015)'den yararlanılabilir.

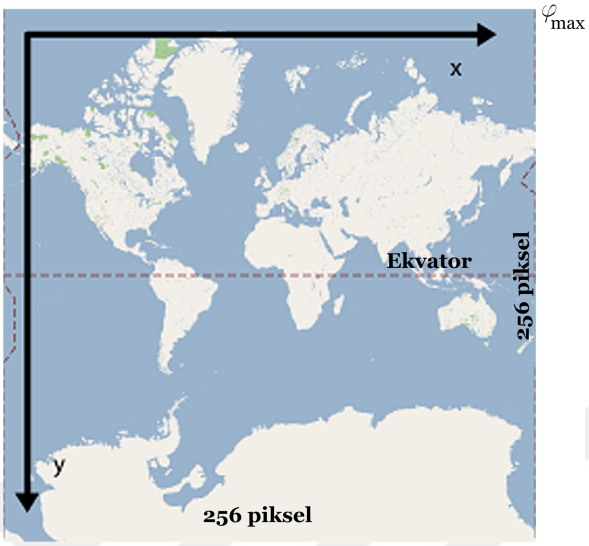
Çevrimiçi haritalarda Web Mercator projeksiyonunun seçimi, özellikle küresel ölçekte, geleneksel Mercator projeksiyonunu hesaplamalar açısından basitleştirmektedir. Haritanın kesintisiz kaydırma, yakınlaştırma işlemleri için hızlı hesaplamalar gerçekleştirilebilmektedir. Aynı zamanda silindirik bir projeksiyon olmasından dolayı kuzey doğrultusu, harita üzerinde zoom ya da gezinti yaparken her

seferinde aynı doğrultuda kalmaktadır. Bu da projeksiyonun tekrar hesaplanmasının önüne geçmektedir.

Mozaik parçalarının numaralandırılmasında genellikle üç temel şablon kullanılmaktadır. Bunlardan ilki Google Maps ve OSM tarafından kullanılan isimlendirmedir. Harita kapsamı -180°, +180° boylamları ve 85°,0529 kuzey ve 85°,0529 güney enlemleri arasındaki bölgeyi kapsamaktadır. Diğer isimlendirmeler TMS (Tile Map Service) ve Microsoft tarafından kullanılan QuadTrees yöntemlerini içermektedir. Google Map API tarafıdan kullanılan birkaç koordinat sisteminden söz etmek mümkündür. Bunlardan ilki Dünya üzerindeki herhangi bir noktayı yegane (unique) olarak tanımlayan enlem ve boylam değerleridir (Google WGS84 datumunu kullanmaktadır). Bir diğeri de bu enlem ve boylam değerlerinin harita üzerinde yegane şekilde tanımlanmasını sağlayan dik koordinatlardır (Google worldCoordinates olarak tanımlamaktadır). Son olarak da belirli bir mozaik parçasının belirli bir zoom seviyesinde, harita üzerinde nereye yerleştirileceğini belirleyen mozaik koordinatlarıdır. Google Web Mercator projeksiyonunda piksel koordinatlarını hesaplamak için 0 zoom seviyesindeki tek mozaik parçasını kullanmıştır. Bu da dik koordinat uzayının (0-256) ve (0-256) aralığında kullanılabilir olduğunu göstermektedir. Koordinatlar mozaik görüntüsünün sol üst köşesinden başlamaktadır ve bu nokta 180° boylam ve 85°,0529 kuzey enlemine karşılık gelmektedir. x koordinatları doğu (sağ) ve y koordinatları da güney (aşağı) yönde artmaktadır (GoogleDevelopers, 2016b) (Şekil 4.8). Haritanın kuzey ve güneyde 85°,0529 enlemlerinde son bulmasının iki nedeni vardır. İlk olarak 256x256 piksellik mozaik sistemini gerçekleştirmek için 0 zoom seviyesindeki tek görüntüye sığan dünya haritası kare biçiminde olmalıdır. İkinci olarak da zaten kutup bölgelerinin 85° kuzey ve güney enlemlerinden sonra gösterilememesidir. Bu değerin hesaplanmasında 0 zoom seviyesindeki mozaik parçası temel alınmıştır. 256x256 piksel boyutuna sahip bu mozaik parçası, harita alanının yüksekliğinin ekvatora eşit olması durumunu doğurmaktadır. Bu durumda hesaplanabilecek en yüksek enlem değeri için, n zoom seviyesini göstermek üzere, n ve y değerleri sıfıra eşitlenerek;

$$\ln \tan\left(\frac{\pi}{4} + \frac{\varphi_{max}}{2}\right) = \pi \tag{4.11}$$

eşitliği elde edilir. Eşitlik çözüldüğünde maksimum enlem değerinin 85°,0529 olabileceği görülmektedir. Yani harita alanı kuzey ve güneyde bu enlemlerde kesilmekte ve 256x256 boyutunda kare bir mozaik elde edilmektedir.



Şekil 4.8 Google Maps piksel koordinat eksenleri (GoogleDevelopers (2016b))

Bu tek mozaik parçası için dik koordinatlarının harita üzerinde temsil ettiği noktalar hesaplandıktan sonra farklı zoom seviyeleri için de piksel koordinatlarının hesaplanması gerekmektedir. Google bunun için;

$$\text{pixelCoordinate} = \text{worldCoordinate} * 2^{\text{zoomLevel}} \tag{4.12}$$

formülünü kullanmaktadır. Bu formülden, her bir artan zoom seviyesi, x ve y yönünde iki kat büyümektedir, yorumu çıkarılabilmektedir. Bu nedenle her bir büyük zoom seviyesi bir öncekine göre dört kat daha yüksek çözünürlüğe sahiptir. Yakınlaştırma düzeyi 1 olan bir haritada 4 adet 256x256 piksel mozaik bulunmaktadır, bu da piksel uzayında 512x512 boyutlu bir görüntü anlamına gelmektedir. 19 zoom seviyesinde piksel koordinat uzayı 0 ve $256*2^{19}$ aralığında bir değere karşılık gelmektedir. 1 pikselin temsil ettiği mekânsal çözünürlük 0 ve 18 zoom seviyeleri arasında 156km ile 60cm arasında değişmektedir. Bu değer;

$$1 \text{ piksel} = \frac{\pi a}{2^{n+7}}\ metre \tag{4.13}$$

formülü ile hesaplanmaktadır (Bildirici, 2015). Burada n değeri 0 ila 18 arasında değişmektedir. Google Maps API belirlenen zoom seviyesine göre bir merkez noktası (viewport) belirlemektedir. Daha sonra bu nokta kullanılarak görüntünün sınırlayıcı dörtgeni[47] hesaplanarak piksel koordinatlarına dönüştürülür. Bu koordinatlar içerisine düşen mozaik parçaları hesaplanır ve sadece bu mozaikler yüklenerek haritanın görüntülenmesi oldukça kolaylaşır. Tablo 4.3, zoom seviyelerine göre eklenmesi gereken mozaik sayılarını özetleyerek, bu işlemin neden bu kadar önemli olduğunu göstermektedir.

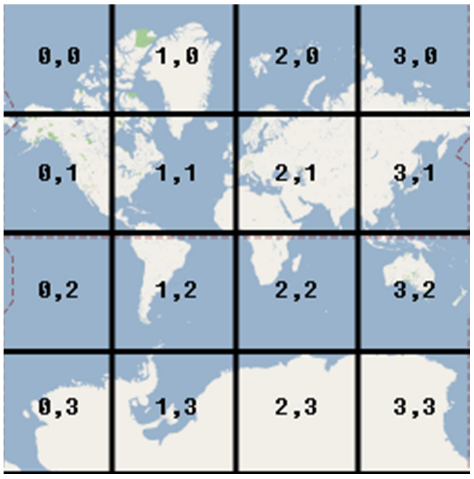
Google Maps API, harita mozaiklerini kullanıcılar harita üzerinde gezinirken ya da yakınlaştırma düzeyini değiştirirken, hangi mozaik parçalarının gerekli olduğunu hesaplayarak yükler. Piksel koordinatlarında olduğu gibi mozaik numaralandırma için de aynı başlangıcı kullanır. Mozaikler haritanın sol üstünden başlayarak, her bir mozaiğe yegane bir (x, y) çifti atanır. x değeri sağa, y değeri aşağı doğru artar (Şekil 4.9).

Tablo 4.3 Zoom seviyelerine göre mozaik sayıları ve boyutları (OSM_wiki, 2016)

Seviye	Mozaik kapsamı	Mozaik sayısı	Mozaik boyutları
0	1 mozaik tüm dünyayı kapsar	1 mozaik	360° x 170.1022°
1	2 × 2 mozaik	4 mozaik	180° x 85.0511°
2	4 × 4 mozaik	16 mozaik	90° x 42.5256°
n	$2^n \times 2^n$ mozaik	2^{2n} mozaik	$360°/2^n$ x $170°.1022/^{2n}$
12	4096 x 4096 mozaik	16 777 216	0.0879° x 0.0415°
16		2^{32} = 4 294 967 296 mozaik	
17		17 179 869 184 mozaik	
18		68 719 476 736 mozaik	
19	Maksimum yakınlık düzeyi	274 877 906 944 mozaik	

Bu değerler kullanılarak ne zaman hangi mozaik parçasının yükleneceği formüle edilebilmektedir. Şekil 4.10, üç farklı yöntem için numaralandırma biçimlerini göstermektedir (MapTiler, 2016).

[47] Bounding Box olarak isimlendirilen sınırlayıcı dörtgen bir şekli minimum boyutlarla içine alan dörtgeni ifade etmektedir.



Şekil 4.9 Google Maps mozaik numaralandırma (GoogleDevelopers (2016b))

4.3.2 Grafik ara yüz tasarımı

İlk harita mashup uygulaması Nisan 2005 tarihinde bir 3 boyutlu grafik sanatçısı olan Paul Rademacher tarafından, emlak verisini o zamanlar yeni ortaya çıkmış Google Maps ile birleştirerek oluşturmuştur (Schmidt ve Weiser, 2012).



Şekil 4.10 Yakınlaştırma düzeylerine göre farklı isimlendirme yöntemleri (MapTiler, 2016)

Bu olay "programlamacı-haritacı" olarak isimlendirilen insanların egemen olduğu yeni bir web haritalama alanı ortaya çıkarmıştır (Plewe, 2007). Web haritalama

servisleri cazip ve sezgisel ara yüzlere taşınmış, API'ler veri mashup uygulamalarına imkân sunmuştur. Ancak bir mashup uygulaması geliştirmek, bir web sayfası oluşturma ve servisi, dağıtımı yapılan veri ile birleştirebilme teknik bilgisine ihtiyaç duymaktadır. Bu da harita tasarımcılarını web geliştirici olmaya zorlamaktadır. Zamanla Google My Maps, Yahoo! Pipes gibi ara yüze sahip harita oluşturucuları ortaya çıkmıştır. Bu ara yüzler kullanıcılara sürükle-bırak mantığında kolay web haritaları oluşturabilme imkânı sunmaktadır. Ancak sadece temel harita oluşturma seçenekleri sunan bu uygulamalar gelişmiş harita mashup uygulamaları oluşturma konusunda oldukça yetersiz kalmaktadır. Bu nedenlerle, tez çalışması için oluşturulan mashup uygulaması için bu gibi yardımcı ara yüzler kullanılmamıştır.

Oluşturulan mashup uygulaması iki ana ara yüzden oluşmaktadır. Birinci ara yüz temel harita ara yüzü olup mashup uygulamasının ana yapısını oluşturmaktadır (Şekil 4.11). İçeriğinde zaman kaydırma aracı (1), karşılaştırma ara yüzüne geçiş (2), gezinti bilgi çubuğu (3), ana menü tuşları (4) ve harita kontrolleri (5) alanlarını barındırmaktadır. Önceki bölümlerde belirtildiği gibi Google Maps API hizmeti kullanarak, web geliştiricileri kendi özel web sayfalarına Google haritalarını ekleyebilme imkânına sahiptir. Google ya da çoğu haritalama servisi herhangi bir ücret talep etmemesine karşın (5. bölümde detaylandırılan bazı sınırlamalar dışında) bu hizmeti bir "API Key" aracılığıyla vermektedir. Kullanıcılar bir Google hesabı açıp bu anahtara kolaylıkla ulaşabilmektedir. Bu işlem sayfanın bir HTML sayfası olduğunun deklerasyonundan hemen sonra HTML dokümanının **`head`** elemanı içerisinde gerçekleştirilir (Şekil 4.12). Bir web sayfasının nasıl tasarlanabileceği, gereksinimleri ve kullanılabilen etiketler hakkında bu tezde fazla teknik detaya yer verilmemiştir. Ancak http://www.w3schools.com/ ve benzeri sitelerde başlangıç düzeyinden gelişmiş düzeye kadar birçok örneğin bulunduğu, çevrimiçi web geliştirme derslerine ulaşmak mümkündür. Burada ilk olarak !DOCTYPE html etiketi ile sayfanın bir HTML5 kodlaması olduğu belirtilmektedir. Bu sayede web tarayıcıları sayfayı HTML5 kurallarına göre derleyerek göstermektedir. Ardından sayfanın içeriği html etiketlerinin içerisine yazılmaktadır. Sayfada görüntülenmeyecek ancak sayfayı düzenlemede kullanılacak özellikler, dış kaynaklar gibi ifadeler head etiketleri içerisinde yer alır. Aynı şekilde sayfanın görüntülenecek kısmında yer alan tüm özellikler de body etiketleri ile sınırlanmaktadır. Bu temel ifadeler bir web sayfasının oluşturulması için yeterlidir ancak yukarıda da ifade edildiği üzere burada fazla teknik detaya girilmeyecektir.



Şekil 4.11 Mashup uygulaması temel harita ara yüzü. 1) zaman kaydırma aracı 2) Karşılaştırma arayüzüne geçiş 3) Gezinti bilgi çubuğu 4) Ana menu tuşları 5) Harita kontrolleri

Web sayfasında Google Maps API servisinin kullanılacağının belirtilmesinden sonra artık istenilen özelliklere sahip harita sayfaya eklenebilmektedir. Bu işlem genellikle **`map-canvas`** tanımlamasıyla oluşturulmaktadır. Şekil 4.12'de verilen örnek kodlamada **`body`** etiketleri içerisinde **`map-canvas`** nesnesinin olduğu görülmektedir. Ancak bu nesne ile ilgili herhangi bir tanımlamaya yer verilmemiştir. İşte bu gibi stil tanımlamaları web sayfasının kodlama içeriğinde bulunabileceği gibi dış bir stil dosyasına da bağlanabilmektedir. Buradaki **`style.css`** dosyası bu amaçla sayfaya eklenmiştir. Şekil 4.13'de verilen örnekte sayfanın ve **`map-canvas`** nesnesinin boyutları tam sayfa boyutunu kaplayacak şekilde belirlenmiştir. Harita özellikleri ise genellikle sayfa yüklendiğinde, sayfayla beraber yüklenmesi amacıyla **`initialize()`** fonksiyonu içerisinde tanımlanmaktadır. Bir HTML sayfa içerisindeki fonksiyonlar JavaScript dili ile yazılmaktadır ve bu kodlamalar da sayfa içerisine **`script`** etiketleri ile eklenebileceği gibi dış bir dosyadan da çağırılabilmektedir.

```
<!DOCTYPE html>
<html>
        <head>
<script type="text/javascript"
src="https://maps.googleapis.com/maps/api/js?key=”API_KEY”&sensor=f
alse">
</script>
<link rel="stylesheet" href="css/style.css">
        </head>
   <body>
    <div id="map-canvas"></div>
```

Şekil 4.12 İçeriğine Google Maps API aracılığıyla bir harita eklenen basit bir HTML sayfası kodlaması.

```
html, body {height: 100%; margin: 0; padding: 0; }
#map-canvas {   width: 100%; height:100%;}
```

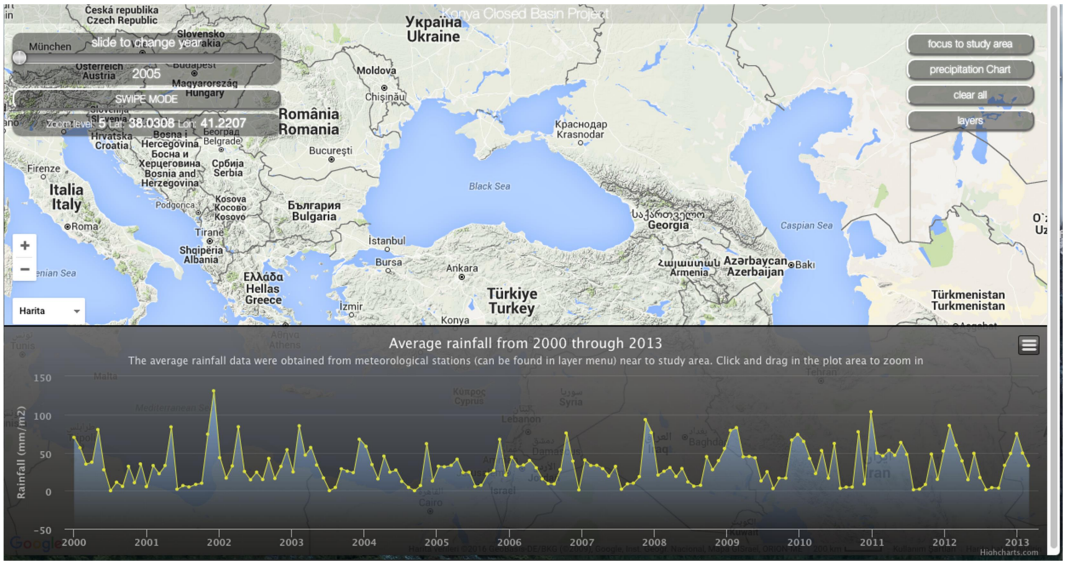
Şekil 4.13 Basit bir biçimlendirme kod parçası

Bunun için stil dosyalarında olduğu gibi başlık kısmında bu JavaScript dosyasının yolu belirtilmelidir. Şekil 4.14 incelendiğinde harita özelliklerinde haritanın yüklenme esnasındaki merkez noktası, yüklenme zoom seviyesi, harita tipi (uydu, arazi ya da yol haritası gibi) ve harita üzerinde bulunması istenen kontrollerin parametreleri görülmektedir. Ve kodun sonunda da sayfaya **`map-canvas`** olarak tanımlanan nesnenin içerisine, **`mapOptions`** parametreleri kullanılarak bir harita yerleştirilmesi istenmektedir. Görüldüğü gibi tüm işlemler HTML, CSS ve JavaScript dilleri ile gerçekleştirilmektedir. Dolayısı ile bu üç teknoloji, web haritaları oluşturma konusunda çalışmak isteyen araştırmacılar için olmazsa olmaz bileşenlerdir. Oluşturulan mashup uygulamasında da buna benzer bir yol izlenmiş ve harita alanından maksimum yararlanma amacıyla tam boyutlu harita tanımlamaları gerçekleştirilmiştir. Tüm menüler ve araçların sadece gerektiğinde ekrana gelmesi sağlanarak harita alanının kısıtlanması engellenmeye çalışılmıştır. Örneğin havzada bulunan meteorolojik istasyonların verilerinden üretilen ve etkileşimli yağış grafiği, hem grafiğin hem de haritanın okunaklığını bozmaması açısından bir tuş ile harekete geçecek şekilde programlanmıştır (Şekil 4.15).

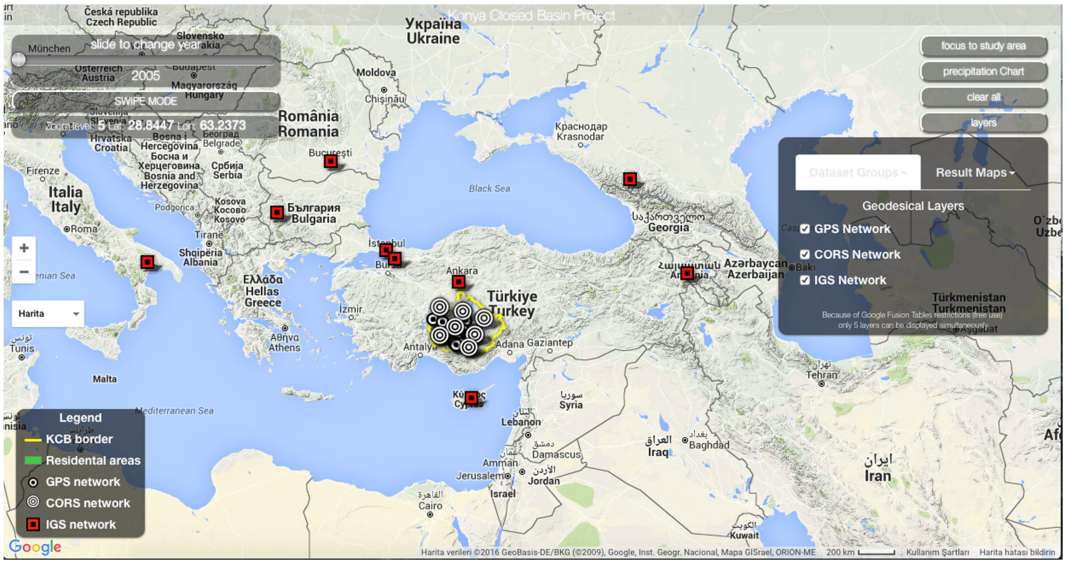
```
function initialize()
                {
                 var mapOptions =
                        {
                center: new google.maps.LatLng(38,  35),
                zoom: 6,
                 mapTypeId: google.maps.MapTypeId.TERRAIN,
                 mapTypeControlOptions:
                 {
               style: google.maps.MapTypeControlStyle.DROPDOWN_MENU,
               position: google.maps.ControlPosition.LEFT_CENTER
                 },
                overviewMapControl: true,
                overviewMapControlOptions: {opened: true},
                scaleControl: true,
                streetViewControl: false,
                panControl: true,
                panControlOptions: {position:
                google.maps.ControlPosition.LEFT_CENTER},
                zoomControl: true,
                zoomControlOptions: {style:
                google.maps.ZoomControlStyle.SMALL, position:
                google.maps.ControlPosition.LEFT_CENTER,}
                        };
map = new google.maps.Map(document.getElementById("map-canvas"),
mapOptions);
                }
```

Şekil 4.14 Harita özelliklerinin başlangıç (initialize) fonksiyonu içerisinde tanımlanması

Aynı şekilde diğer menü ve özellikler de sadece görünmesi gereken zamanlarda ortaya çıkacak şekilde programlanmıştır. Örneğin katman menüsü tuşuna basıldığında katman menusunun açılması ve herhangi bir katmanın ekrana eklenmesi ile lejant penceresinin aktif olması gibi (Şekil 4.16) örnekler verilebilmektedir.

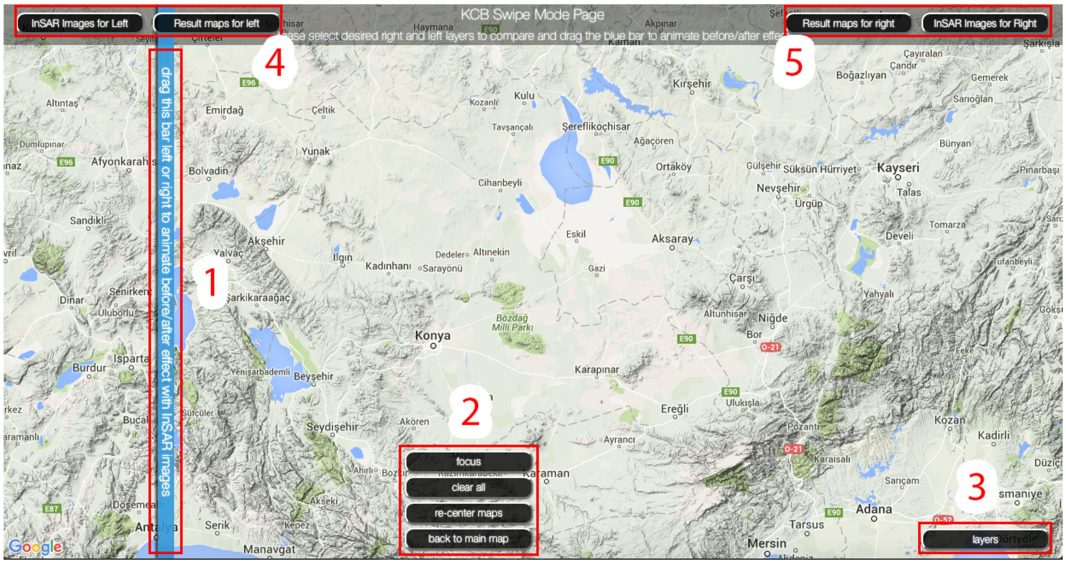


Şekil 4.15 Dinamik yağış grafiğinin aktif hali



Şekil 4.16 Katman menusunun aktif hali ve eklenen katmanların aktifleştirdiği lejant penceresi

İkinci ara yüz ise iki farklı katmanın karşılaştırılmasına imkân verecek şekilde tasarlanmıştır (Şekil 4.17). Sayfanın içeriğinde bir görüntü süpürme çubuğu (1), menü tuşları (2), katman menüsünü aktifleştiren katmanlar tuşu (3) ve süpürme aracının sağında ve solunda kalan alanlar için karşılaştırma katmanları menüleri (4)-(5) yer almaktadır. Detayları sonraki bölümlerde (bölüm 4.3.7) verilecek olan ara yüz ile kullanıcılar bir süpürme çubuğunun sağına ve soluna farklı katmanlar ekleyebilme imkânına sahiptir. Bu sayede farklı iki katman arasındaki farklar görsel olarak izlenebilmektedir.



Şekil 4.17 Katman karşılaştırma ara yüzü

4.3.3 Veri katmanları

Çalışmada kullanılacak verilerin depolanması, yönetimi ve harita üzerine katman olarak eklenebilmesi amacıyla Google Fusion Tables (ya da kısaca Fusion Tables) servisi kullanılmıştır. Fusion Tables bulut tabanlı bir veri yönetim ve entegrasyon servisidir. Kullanıcılara tablo şeklindeki dosyaları (hesap tabloları, CSV, KML) 100MB boyutuna kadar yükleyebilme imkânı sunmaktadır. Sistem, veri görselleştirme için birçok yol kullanmaktadır (grafikler, haritalar ve zaman çizelgeleri gibi) ve bu verileri filtreleme ve birleştirme işlemleri gerçekleştirilebilmektedir. Farklı kullanıcılara ait çoklu kaynaklardan alınan verileri tablolar arasında ilişkiler kurarak entegre edebilmektedir. Kullanıcılar verilerini gizli tutabilir ya da kullanıma açarak diğer kullanıcıların kullanmasına izin verebilir. Bu sayede veri tabloları arama motorlarının sonuçlarında görüntülenebilmektedir (Gonzalez ve ark., 2010). Verilerin depolanması için yine Google tarafından sağlanan bulut tabanlı teknoloji Google Drive kullanılmıştır. Bu işlem kişisel verilerin saklanması için kullanıcı adı ve parola ile girilebilen ve veri paylaşımının kısıtlanabildiği güvenli bir ortam sunmakta ve verilerin özel sunucularda tutulması zorunluluğunu da ortadan kaldırmaktadır.

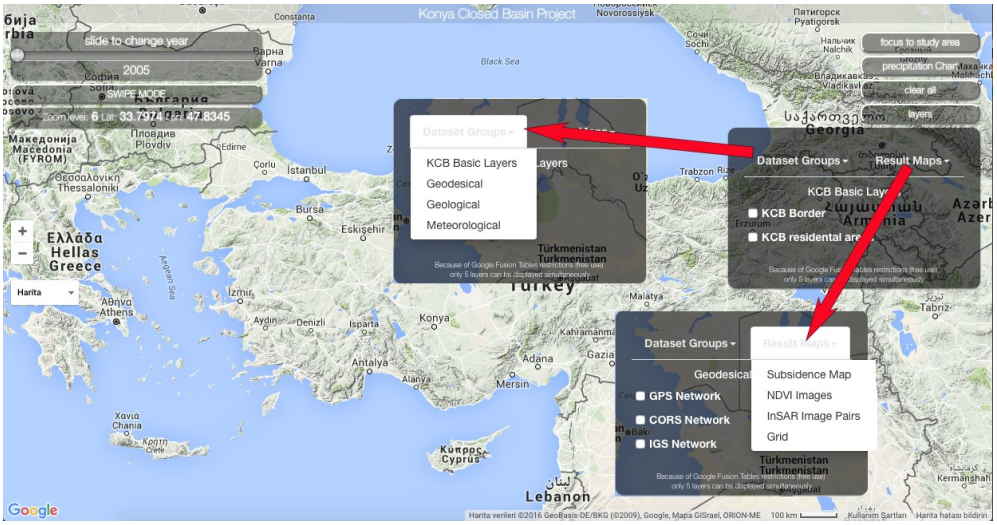
Tablosal yapıda düzenlenmesi uygun olan veriler Fusion Tables servisine aktarılmıştır. Yukarıda da belirtildiği gibi Fusion Tables XLS, CSV (Comma Separated Values) ve KML formatlarına tam destek sunmaktadır. Dosya bir kez sisteme aktarıldığında her bir dosya için benzersiz bir ID numarası atanmakta ve bu numaralar tablonun farklı ortamlara çağırılmasında kullanılmaktadır. Ayrıca hali hazırda bir Google servisi olan Fusion Tables tablolarının katman olarak Google haritalar üzerine eklenme desteği de bulunmaktadır. Kullanıcılar basit bir kodla bir Fusion Tables katmanı üretebilmekte ve bunu harita üzerine ekleyebilmektedir. Şekil 4.18'de bir örnek Fusion Tables Layer kodlaması verilmektedir. Görüldüğü gibi tablonun herhangi bir kolonundan belirlenen koşula göre sorgulama yapıp sadece bu sorgu sonucu eklenebilmektedir. Bu çözüm, az sayıda katmana sahip uygulamalarda ya da tüm katmanların harita yüklenmesi aşamasında haritayla birlikte eklenmek istendiği durumlarda kullanışlıdır. Ancak bu çalışmada da olduğu çok sayıda katmanın yönetilmesi ve katmanların tercihlere göre harita üzerine eklenmesi istenen uygulamalarda farklı yaklaşımlara ihtiyaç duyulmaktadır. Bu problemin çözümü için geliştirilen yaklaşımda öncelikle katmanların HTML div elemanları ile gruplandırılması ve daha sonra JavaScript dilinin bir kütüphanesi olan jQuery kullanarak HTML input

elemanı ile kullanıcı tercihine bağlı katman ekleme seçeneği sunulmaktadır. Öncelikle JavaScript dosyasına tüm katmanlar Fusion Tables Layer olarak tanımlanmıştır.

```
var layer = new google.maps.FusionTablesLayer
        ({
        map: map,
        query:
                {
        select: "col2",
        from: "Tablo ID Numarası",
        where: ""
                },
         });
Layer.setMap(map);
```

Şekil 4.18 Örnek Fusion Tables Layer kodlaması

Daha sonra her bir katman için **`input`** elemanları oluşturulmuş ve tür olarak da **`checkbox`** işaret kutuları atanmıştır (Şekil 4.19). Kullanıcı herhangi bir katmanı işaretlediğinde o işaret kutusuna atanan katman Fusion Tables sisteminden ID numarası ile sorgulanmakta ve harita üzerine eklenmektedir. Burada jQuery kütüphanesinin görevi katmanların açılıp kapanmasını sağlamaktır. Katman gruplandırmada ise CSS ve HTML özellikleri kullanılmıştır. Her katman grubu için oluşturulan alanlar yine işaretlenen katman grubu başlığı altında jQuery ile açılıp kapanmaktadır. Şekil 4.19'de katman grupları ve içerikleri gösterilmektedir. Geliştirilen bu yaklaşım sayesinde hem istenen katman istenen zamanda harita üzerine eklenebilmekte hem de katman menülerinin dinamik olarak hareket etmesiyle harita alanı kısıtlanmamış olmaktadır.



Şekil 4.19 Katman grubu menüleri ve içerikleri

```
<div id="layersDiv" class="layersDiv">
 <div class="bs-example">
   <ul class="nav nav-tabs" id="myTab">
     <li class="dropdown">
         <a data-toggle="dropdown" class="dropdown-toggle"
           href="#" style="color: #E9E9E9">Dataset Groups
           <b class="caret">
           </b>
          </a>
         <ul class="dropdown-menu">
             <li><a href="#kcb-basic">KCB Basic Layers</a></li>
             <li><a href="#geodesical">Geodesical</a></li>
             <li><a href="#geological">Geological</a></li>
             <li><a href="#meteorological">Meteorological</a></li>
         </ul>
     </li>
<div class="tab-content">
   <div id="kcb-basic" class="tab-pane fade in active">
      <h5>KCB Basic Layers</h5>
          <div class="cb-row">
             <input type="checkbox" class="form-checkbox"
              name="checkbox-1"  id="checkbox-1"
              onchange="toggleLayer(layer1)" />
              <label for="checkbox-1">KCB Border</label>
     <script type="text/javascript">
        document.getElementById("checkbox-1").checked = false;
     </script>
</div>
```

Şekil 4.20 HTML dokümanı içerisindeki <input> elemanları ve katman grupları

Bu işlemlerin otomatik olarak yapılabileceği herhangi bir araç bulunmamaktadır. Daha önce de belirtildiği üzere tez kapsamında geliştirilen bu tür uygulamalar uyarlamalı uygulamalar olup geliştirilen özellikler harita üreticisinin ihtiyaçları, programlama düzeyi ve hayal gücü ile sınırlıdır. Tüm katman gruplarının Fusion Tables Layer ile desteklenmemesi durumunda alternatif çözümler üretilmelidir. Bu noktada da kullanılan harita servisinin dış veri ile entegrasyon yetenekleri devreye girmektedir. Örneğin görüntü dosyalarının harita üzerinde görüntülenebilmesi için Google Maps API görüntü bindirme desteği sunmaktadır. Fusion Tables servisinin henüz görüntü bindirme özelliği bulunmamaktadır[48]. Şekil 4.21'de örneklendiği gibi sınırları coğrafi koordinatlarla belirli görüntüler harita üzerine katman olarak eklenebilmektedir.

Fusion Tables katmanlarında olduğu gibi bu katman da JavaScript dosyasının içerisinde tanımlanarak daha sonra jQuery ile çağırılabilmektedir.

[48] Google Fusion Tables özellikleri hakkında detaylı bilgi https://support.google.com/fusiontables/answer/2571232 adresinden edinilebilir.

```
var imageBounds = new google.maps.LatLngBounds
    (
      new google.maps.LatLng(37, 32.16667),
      new google.maps.LatLng(38.1666667, 33.66667)
    );
var overlayOpts = {opacity:1.0}
var layer = new google.maps.GroundOverlay(
    'images/subsidence_map.png',imageBounds,overlayOpts );
```

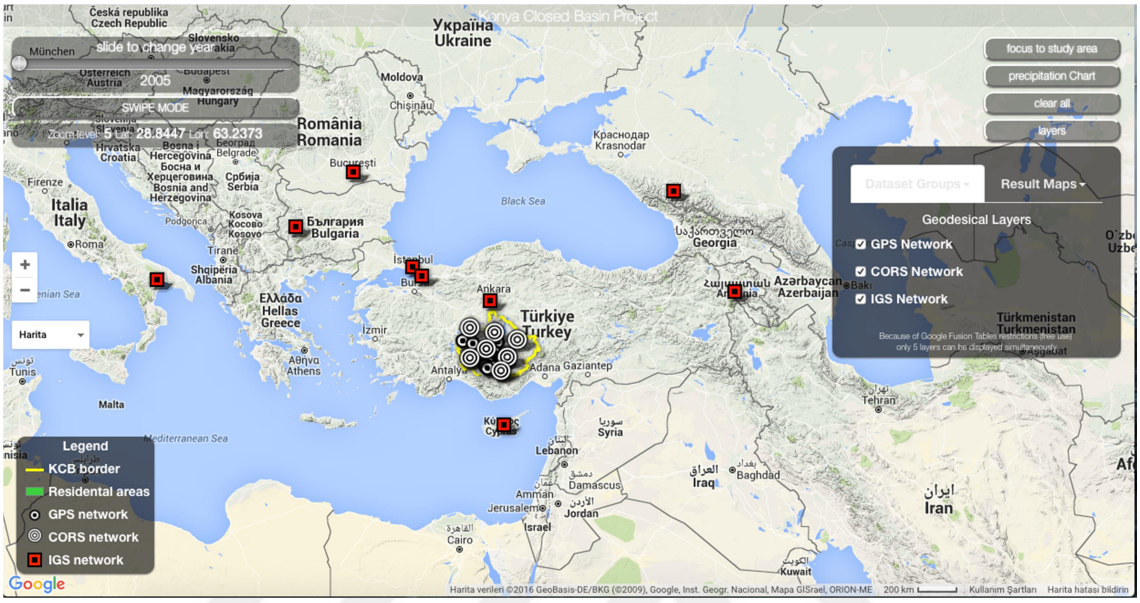
Şekil 4.21 Google Maps API görüntü bindirme kodlaması örneği

4.3.4 Dinamik lejant yaklaşımı

Harita mashup uygulamalarında, harita üzerine farklı veri kaynaklarından eklenen nesnelerin özel işaretlerinin yer aldığı lejantların oluşturulması karmaşık bir konudur. Aslında basit uygulamalar için geliştirilmiş yöntemler mevcuttur. Örneğin tek bir veri grubu ya da aynı geometriden (nokta, çizgi, alan) oluşan veri setleri için otomatik yöntemler bulmak mümkündür. Örneğin Fusion Tables servisi tek bir tablonun ya da aynı geometriden farklı tabloyla birleştirilmiş verinin lejant seçeneğini sunmaktadır. Ancak bu durum veri setinin faklı geometrilerden oluşması durumunda kullanılamamaktadır. Tek bir geometrinin kullanıldığı durumlarda da oluşturulan lejant oldukça kısıtlı özelleştirme seçenekleri sunmaktadır.

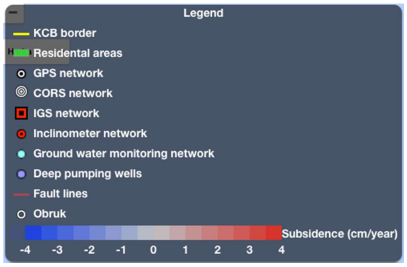
Bu tezde bu probleme yeni bir yaklaşım geliştirilmeye çalışılmıştır. Öncelikle tüm veri seti için kullanılan özel işaretler belirlenerek bir özel işaret kitaplığı oluşturulmuştur. Kitaplığın içeriği ".png" uzantılı resim dosyalarından oluşmaktadır. Harita üzerinde kullanılan her bir nesneye ait özel işaretlerin resim dosyaları oluşturularak daha sonra nesnenin bulunduğu katman haritaya eklendiğinde lejant penceresine eklenecek şekilde programlanmıştır. jQuery yetenekleri kullanılarak lejant penceresinin de sadece o an harita üzerine eklenen katmanların işaretlerini görüntülemesi sağlanmıştır. Çünkü oluşturulan uygulamalar dinamik uygulamalar olup statik haritalarda kullanılan lejant alanı mantığı bu gibi uygulamalarda kullanılamamaktadır. Statik ya da baskı haritalarda harita üzerinde görüntülenmek istenen tüm nesneler hali hazırda mevcut olup lejant alanında da yer alması istenen özel işaret karşılıkları yer almaktadır. Dinamik haritalarda o an var olmayan bir nesnenin lejant alanında özel işaretinin görüntülenmesi kullanıcının harita ile iletişiminde problemlere neden olacaktır. Buradaki çalışma mantığı katman menüsü ile benzer

niteliktedir. Aradaki fark lejant penceresinin açılmasını sağlayacak bir tuşun bulunmamasıdır. Herhangi bir katman haritaya eklendiğinde o katmanın işareti ve etiketiyle aktif hale gelmektedir ve tüm katmanlar kapatıldığında lejant penceresi de görüntüden çıkarılmaktadır (Şekil 4.22).



Şekil 4.22 Eklenen katmanlara göre aktif hale gelen dinamik lejant penceresi

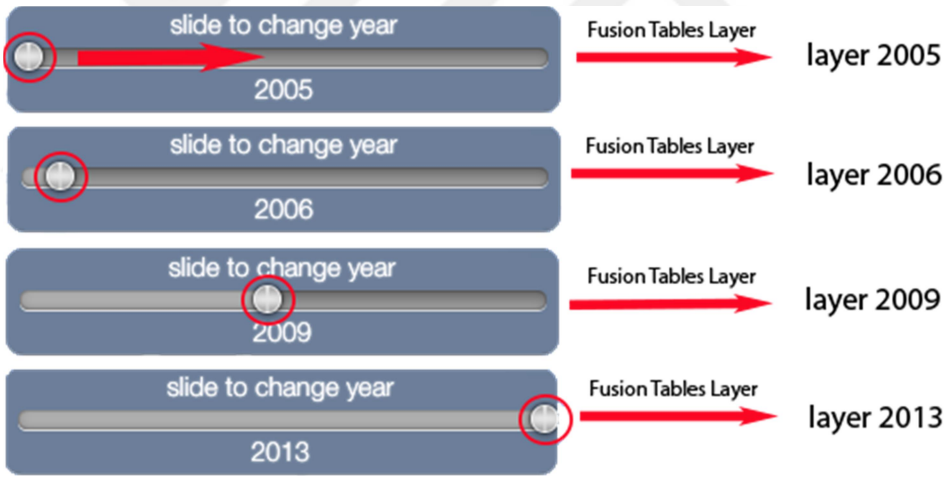
Bu işlem teknik olarak iç içe sıralanan **`div`** elemanları ile gerçekleşmektedir. Kullanıcı bir katmanı aktif hale getirdiğinde, ilgili imaj dosyası kitaplıktan çağırılarak **`div`** alanı içerisine yerleştirilmekte ve işaretin etiketi de bu imajın yanına yazılmaktadır. Özel işaretlerin lejant penceresindeki sıralanışı, katman ekleme sırasına göre gerçekleşmektedir. Görüntü bindirme katmanları için kullanılan yaklaşımda ise renk aralıkları ve aralık değerleri kullanılarak bir **`div`** dizisi oluşturulmuştur. Her bir dizi elemanı için boyut ve dolgu renkleri belirlenerek lejant oluşturulmuştur. Şekil 4.23'de nokta, çizgi, alan ve görüntü bindirme nesneleri için üretilmiş özel işaretler gösterilmektedir.



Şekil 4.23 Lejant penceresi içeriği

4.3.5 Zaman kaydırma aracı

Veri seti içerisinde bulunan zamana bağlı katmanların görselleştirilmesi için bu çalışmada zaman kaydırma aracı yaklaşımı kullanılmıştır. Bu yaklaşımda aynı veri setinin farklı zamanlara ait katmanları bir kaydırma çubuğunun hareketiyle harita üzerine çağırılmaktadır. Bu sayede verinin zamana bağlı değişimi canlandırılmış olmaktadır. Şekil 4.24'de zaman kaydırma aracının çalışma prensibi örneklenmeye çalışılmıştır. Buradaki süreç kullanıcı kaydırma çubuğunu hareket ettirdiğinde belirlenen aralıklara ulaştığında JavaScript kodu içerisinde bir sorgu oluşturmaktır. Daha önce tanımlanan katmanlar jQuery aracılığıyla çağırılmakta ve bir önceki katman ekrandan temizlenerek eklenmektedir. Şekil 4.25'de verilen örnekte 2005 tarihli katman varsayılan olarak ayarlanarak tarih ilerletildiğinde sırasıyla 2006 ve 2007 katmanlarının haritaya eklenmesi sağlanmaktadır. **`clearmap()`** fonksiyonu ile de bir önceki katman haritadan kaldırılmakta ve aynı anda iki katmanın açılması önlenmektedir.



Şekil 4.24 Zaman kaydırma aracı

```
function changeLayer(tableidselections)
{
  if (tableidselections == 2005){clearmap();}
  if (tableidselections == 2006){clearmap();layer7.setMap(map); }
  if (tableidselections == 2007){clearmap();layer8.setMap(map); }
}
function clearmap() {layer7.setMap(null);layer8.setMap(null);}
```

Şekil 4.25 Zaman kaydırma aracı örnek kodlaması

4.3.6 İstatistiksel yüzey (izoplet harita) yaklaşımı

Şimdiye kadar kullanılan veri türleri (nokta, çizgi, alan, görüntü bindirme) için web harita servislerinin sunduğu olanaklardan yararlanılmıştır. Örneğin nokta, çizgi ve alan nesnelerinin konuma bağlı görselleştirilmesi ve veri setinde bulunan herhangi bir özniteliğe göre tematik haritalarının üretilmesi konusunda Fusion Tables çözümler sunmaktadır. Aynı zamanda Google Maps API tarafından sunulan görüntü bindirme desteği ile görüntü dosyalarının da katman olarak eklenebileceği önceki bölümlerde incelenmiştir. Bunlara ek olarak Fusion Tables servisi "heat maps" olarak isimlendirilen haritaların oluşturulmasını da desteklemektedir. Bu özellik, kullanıcıların harita üzerindeki nesnelerin koordinat uzayındaki yoğunluğunu renklendirilmiş olarak görüntülenmesini istediğinde, oldukça kullanışlıdır. Aynı zamanda çok büyük boyutlu veri setlerinde nesne yoğunluğundan dolayı okunaklığın azalmasını engellemek için de kullanılabilmektedir. Fusion Tables mimarisinde şu şekilde gerçekleştirilmektedir:

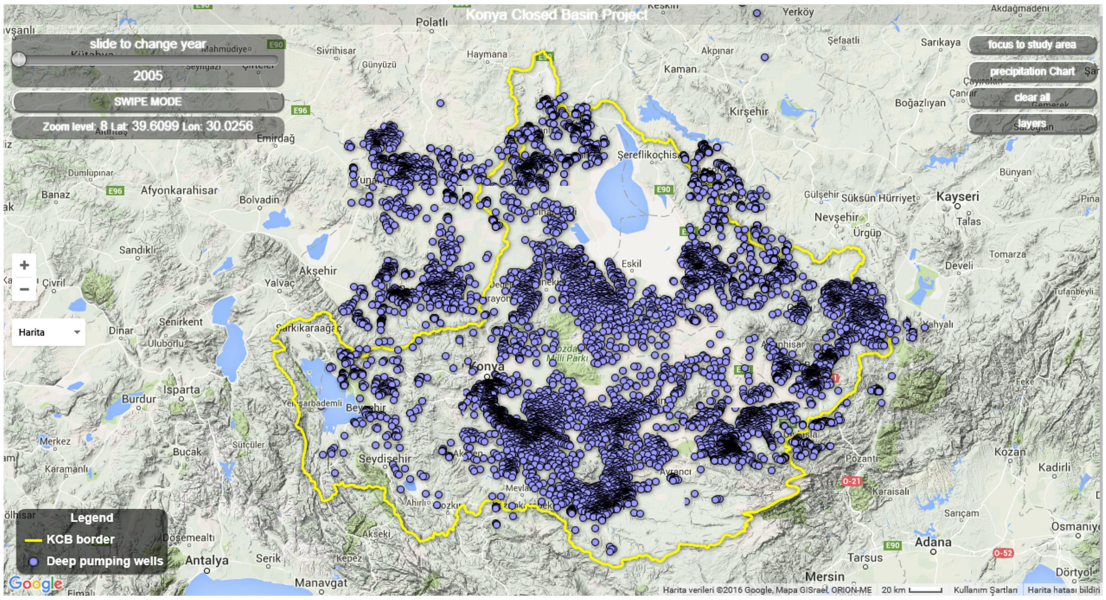
(i) Öncelikle görüş alanındaki nesneler mekânsal indeksleri kullanılarak belirlenmektedir,

(ii) Görüş alanı küçük gridlere bölünmektedir,

(iii) Her bir grid hücresi içerisine düşen nesne sayısı hesaplanmaktadır,

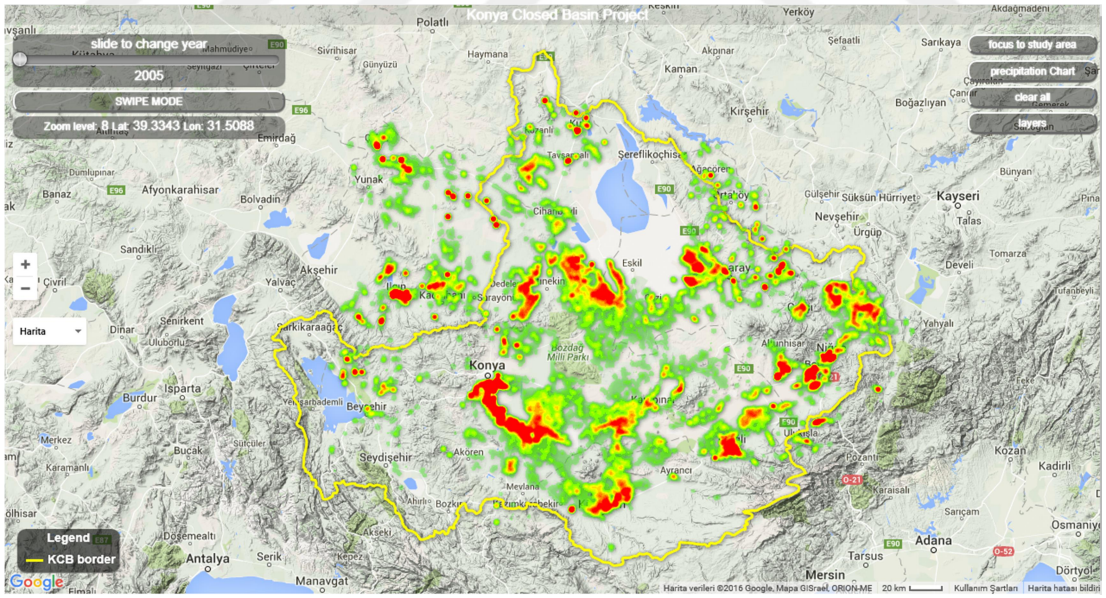
(iv) Grid hücreleri düşük sayılı hücrelere açık, yüksek sayılı hücrelere de koyu renklerin atanacağı bir renk paleti kullanılarak renklendirilmektedir.

İçerisinde nesne bulunmayan hücreler renklendirilmemektedir (Gonzalez ve ark., 2010). Bu yöntem çalışma bölgesi içerisindeki sondaj kuyularının görselleştirilmesinde kullanılmıştır. Çalışma bölgesi geniş bir alana sahip olmasına karşın sayıları yaklaşık 100.000 adete ulaşan kuyuların gösterimi için oldukça kısıtlı bir alandır. Özellikle görüntüleme ölçeği küçüldükçe harita okunaklılığı oldukça düşmekte, neredeyse noktalar bölgeyi tamamen kapatmaktadır (Şekil 4.26). Çalışmada bu probleme çözüm getirmek amacıyla, hâlihazırda Fusion Tables tablosu olarak depolanan sondaj kuyuları veri tablosunun, heatmap özelliği kullanılmıştır (Şekil 4.27).



Şekil 4.26 Çalışma bölgesinde bulunan sulama amaçlı sondaj kuyuları (~100.000 adet)

Herhangi bir tablonun heatmap özelliğinin aktif hale getirilebilmesi için katman olarak tanımlanan tabloların bu özelliğinin aktifleştirilmesi gerekmektedir. Şekil 4.28'de herhangi bir Fusion Tables katmanının heatmap özelliğinin nasıl aktifleştirilebileceği konusunda örnek bir kodlama gösterilmektedir.



Şekil 4.27 Çalışma bölgesinde bulunan sulama amaçlı sondaj kuyularının heatmap gösterimi

Tüm bu görselleştirme avantajlarının yanında web haritalama servislerinin yetersiz kaldığı noktalar da vardır. Özellikle tematik haritaların otomatik üretiminde sadece sınırlı özellikler desteklenmektedir.

```
layer = new google.maps.FusionTablesLayer
                        map: map,
                        heatmap: { enabled: true },
                        query:{
                        select: "col2",
                        from: "TableID",
                        where: ""
                        },
                });
```

Şekil 4.28 Fusion Tables katmanlarının heatmap özelliğinin aktifleştirilmesi kodlama örneği

Genellikle koroplet, orantılı işaretler, nokta yoğunluğu, pasta ve bar grafiği gibi konum bilgisi içeren bir nesneyle ilişkilendirilebilen[49] temel tematik haritalar dışında, ara değerleri belirli enterpolasyon yöntemleriyle belirlenen istatistiksel tematik katmanlar (örn., eşdeğer haritaları, izoplet haritalar gibi) desteklenmemektedir. Web harita üreticileri, bu gibi tematik haritalar üretmek istediğinde ya dış bir yazılımla tematik katmanları üretip görüntü olarak dışa aktararak, web haritaları üzerine görüntü bindirme olarak eklemek, ya da kendi çözümlerini (en azından bu tezin yayınlandığı tarihlerde) geliştirmek zorunda kalmaktadır. Bu çalışmada da bu gibi tematik katmanlara ihtiyaç duyulmuş ve probleme yeni bir yaklaşımla çözüm getirilmeye çalışılmıştır.

Konuyu detaylandırmak için katman kavramı ve Google Maps API tarafından desteklenen katman türlerinin açıklanması gerekmektedir. Katmanlar, harita üzerinde bir ya da daha fazla ayrı parçadan oluşan ancak tek bir birimmiş gibi kontrol edilen nesnelerdir. Genellikle ortak bir ilişkinin tanımlanması için harita üzerine eklenen nesne koleksiyonlarıdır. Google Maps API bu katmanların harita üzerinde tek bir birim olarak yorumlanmasını ve haritanın değişen gösterim seviyelerine göre görüntülenmelerini yönetmektedir.

Google Maps API birkaç katman türünü desteklemektedir:

- Google Maps Data Layer kişisel mekânsal veri için bir yer tutucu (container) sağlamaktadır. Data Layer özelliği kullanılarak özel veriler depolanabilir ya da GeoJSON verisi Google Maps üzerinde görüntülenebilir.
- Fusion Tables layer, Google Fusion Tables bünyesinde depolanan veriyi yorumlayabilmektedir.

[49] Burada ifade edilmek istenen konu nokta ve alan gibi nesnelerin verisi ile bu nesnelerin belirli konumlarını (merkez noktası, ağırlık merkezi gibi) temel işaretleştirmelerdir.

- Heatmap layer coğrafi veriyi "Heatmap" gösterimi olarak yorumlayabilmektedir.
- KML Layer, KML ve GeoRSS elemanlarını bir Maps API V3 mozaik olarak yorumlayabilmektedir.
- Maps Engine Layers, Google Maps Engine tarafından sağlanan verinin kullanıcıların kendi haritalarına eklenmesini kolaylaştıran bir özelliktir. Maps Engine layer mozaik görüntüleri sunucu tarafında oluşturmakta ve istemcilere ulaştırmaktadır. Dinamik Maps Engine katmanı, istemci tarafında yeniden biçimlendirmeye olanak sunarak vektör veri döndürebilmektedir. Bu katman özelliği Google tarafından durdurulmuştur ve kullanımı 29 Ocak 2016 tarihinde son bulmuştur.
- Traffic Layer harita üzerinde trafik durumunu görüntülemeye olanak sağlamaktadır.
- Transit Layer şehirlere ait yerel ulaşım ağının harita üzerinde görüntülenmesini sağlamaktadır.
- Bicycling Layer bisiklet patikalarının harita üzerinde görüntülenmesi ya da bisiklet odaklı özelliklerin ortak bir katman üzerine eklenmesini sağlamaktadır. DirectionRenderer özelliği içerisinde de seyahat modunun bisiklet seçilmesi halinde varsayılan olarak döndürülmektedir.

Bu bilgiler dikkate alınarak çalışma kapsamında oluşturulması planlanan tematik katmanların Data Layer özelliği kullanılarak oluşturulmasına karar verilmiştir. Kullanılan veri nokta, çizgi ya da alan gibi geometriler içermekteyse, API bu geometrileri varsayılan noktalar, çizgiler ve alanlar gibi yorumlayabilmektedir. Kullanıcılar bu nesneleri normal bir katmanda olduğu gibi ya da veri seti içerisinde yer alan özelliklere dayalı biçimlendirme kurallarına göre biçimlendirebilmektedir. **`google.maps.Data`** sınıfı kullanıcılara, haritalarına GeoJSON verisi ekleme imkanı da sunmaktadır. GeoJSON, daha önce de belirtildiği gibi internet üzerinde coğrafi veriler için oluşturulmuş bir standarttır. Data sınıfı GeoJSON yapısını tanır ve GeoJSON verisini görüntülemeyi kolaylaştırır. Bunun yanında **`google.maps.Data`** sınıfı kullanılarak harici veriler modellenebilmektedir.

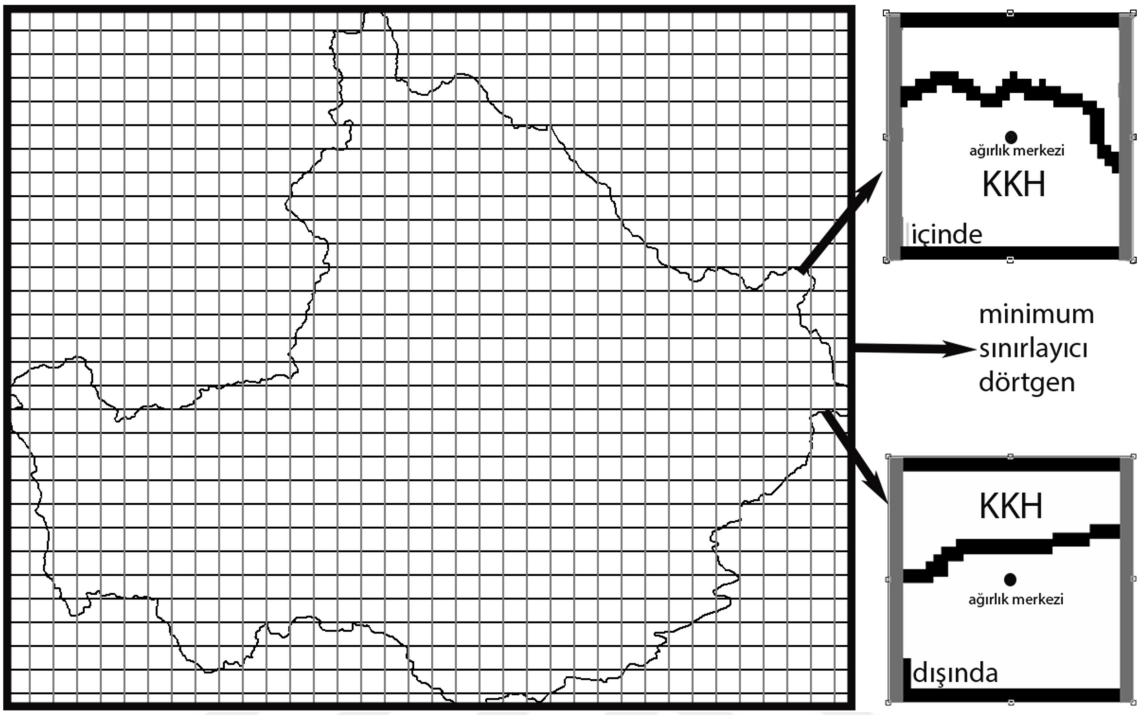
4.3.6.1 Point-in-Poligon testi

Çalışmada oluşturulması planlanan tematik katmanlar, mevcut verilerden türetilerek oluşturulacak istatistiksel yüzeylerdir. Bu amaca ulaşabilmek için öncelikle çalışma bölgesinin, belirli bir çözünürlükte grid alanlarına bölünmesi ve daha sonra Google Maps API kullanılarak harita üzerine katman olarak eklenebilmesi için uygun formata dönüştürülmesi gerekmektedir. Önceki bölümlerde verilen bilgilerden hareketle elde bulunan seçeneklerin KML ya da GeoJSON formatları olduğu görülmektedir. Tematik katmanların oluşturulmasında izlenen işlem adımları aşağıda açıklanmıştır.

Çalışma bölgesinin belirlenen büyüklükte grid hücrelerine ayrılması işlemi için referans olarak Konya Kapalı Havzası sınır çizgisini oluşturan nokta kümesinin KML formatındaki dosyası kullanılmıştır. Algoritma C programlama dilinde geliştirilmiştir. Program öncelikle KML formatındaki çalışma bölgesi sınırına ait köşe noktalarını okumakta ve bu bölgeyi sınırlandıran minimum sınır dörtgenini (burada box olarak anılacaktır) oluşturmaktadır. Ardından, belirlenen boyutta kare şeklindeki hücreler oluşturulmaktadır (Şekil 4.29). Hücreler herhangi bir sınırlama tanımlanmadığı takdirde box sınırlarını kapsayacak miktarda üretilmektedir. İlgilenilen alan dışında istenmeyen hücrelerin elimine edilmesi gerekmektedir. Hücrelerin havza sınırı dışında kalanlarını ayıklamak için "Point in Polygon" testi kullanılmıştır. Bu testte her bir hücrenin merkez noktasının havza sınırı içinde olup olmadığına bakılmakta ve havza sınırı dışındaki merkez noktalarına sahip olan hücreler elimine edilmektedir.

Point in polygon testi, poligonu (alansal objeyi) oluşturan kenarlar ile test edilen nokta ve poligonun dışındaki bir noktanın oluşturduğu doğru parçasının (test doğru parçası) kesişimine dayanmaktadır. Test doğru parçası ile poligonun kesişim noktalarının sayısı tek ise noktanın poligonun içinde olduğuna, değilse noktanın poligonun dışında olduğuna karar verilmektedir. Poligonun şekli ne kadar karmaşık olursa olsun, bu basit test yöntemi doğru sonuç vermektedir. Bu işlem iki boyutlu koordinat uzayında, iki doğru parçasının kesişiminin belirlenmesi ve kesişim sayısının hesaplanması ile gerçekleştirilmektedir. Burada gerçekleştirilen testte sadece kesişimin var olup olmadığına ve kesişim sayısına bakılmaktadır. Kesişim noktasının koordinatları bu test için gerekli olmadığından hesaplanmamıştır. Test edilen şekillerin karmaşıklığı ve kenar sayısı, hesaplamaların sayısını arttırabilmektedir. Örneğin bu çalışmada KKH sınır çizgisi ile oluşturulan grid hücreleri test edilmiştir. Sınırın yaklaşık 1400 adet doğru parçasından oluştuğu düşünüldüğünde her bir hücre için 1400

kez keşişim hesaplanması gerekmektedir. Hesaplamaları hızlandırmak amacıyla box testi olarak adlandırılan bir programlama tekniğinden yararlanılmıştır.



Şekil 4.29 Minimum sınırlayıcı dörtgen ve hücre merkez noktalarının sınır içinde/dışında bulunma durumu

Geometrik olarak iki doğru birbirine paralel değilse kesişir. Ancak CBS, bilgisayar destekli çizim ve genel olarak grafik programlamada iki doğrunun değil iki doğru parçasının kesişimi önemlidir. İki doğrunun kesişiminin genel ifadesi, birinci doğru,

$$Ax + By + C = 0 \tag{4.14}$$

ikinci doğru,

$$Ex + Fy + G = 0 \tag{4.15}$$

olmak üzere,

$$\begin{bmatrix} X_s \\ Y_s \end{bmatrix} = \begin{bmatrix} (GB - FG)/(FA - EB) \\ (CE - AG)/(FA - EB) \end{bmatrix} \tag{4.16}$$

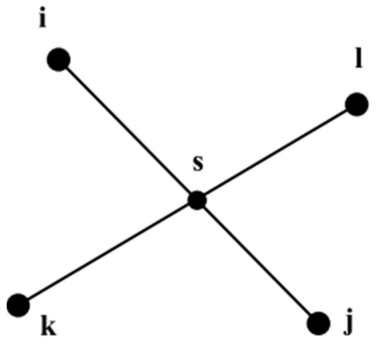
ile gösterilir. Burada X_s ve Y_s kesişim noktalarının koordinatlarıdır. İki doğru paralel ise,

$$FA - EB = 0 \tag{4.17}$$

olur. Bu durumda kesişim mevcut değildir. Yukarıdaki gibi lineer denklem formu ile çözüm yapılırsa önce kesişim noktası hesaplanır, sonra kesişim noktasının iki doğrunun üzerinde olup olmadığı araştırılır. Çeşitli özel durumların dikkate alınmasını gerektiren bu yaklaşım yerine, programlama açısından daha kullanışlı olan ve nokta koordinatlarına dayanan aşağıdaki yöntem izlenebilir. Birinci doğru parçasının başlangıç ve bitim noktaları i(x_i, y_i) ve j(x_j, y_j), ikinci doğru parçasının başlangıç ve bitim noktaları k(x_k, y_k) ve l(x_l, y_l) ile ifade edilirse (Şekil 4.30),
Doğru parçalarının paralel olup olmadığını belirlemek için d parametresi hesaplanır:

$$d = \big(x_j - x_i\big)(y_k - y_l) - (x_k - x_l)(y_j - y_i) \tag{4.18}$$

$d = 0$ ise doğru parçaları paralel olup kesişim hesaplanamaz. $d \neq 0$ ise p_1 ve p_2 parametreleri hesaplanır.



Şekil 4.30 İki doğru parçasının kesişimi

$$p_1 = \frac{(x_k - x_l)(y_i - y_k) - (x_i - x_k)(y_k - y_l)}{d} \tag{4.19}$$

$$p_2 = \frac{(x_i - x_k)\big(y_j - y_i\big) - (x_j - x_i)(y_i - y_k)}{d} \tag{4.20}$$

$0 \leq p_1 \leq 1$ ve $0 \leq p_2 \leq 1$ eşitsizlikleri sağlanıyor ise kesişim noktası doğru parçalarının üzerindedir. Kesişim noktasının koordinatları,

$$x_s = x_i + p_1(x_j - x_i) \quad (4.21)$$

$$y_s = y_i + p_1(y_j - y_i) \quad (4.22)$$

şeklinde hesaplanmaktadır. Bu çalışmada, daha önce de belirtildiği üzere sadece kesişimin var olup olmadığı üzerinde durulmuştur. Bu durumda önce paralellik incelenmiş, daha sonra p_1 ve p_2 parametreleri hesaplanarak kesişimin doğru parçaları üzerinde olup olmadığı belirlenmiştir. Point in polygon testi için bu işlem yeterlidir.

p_1 ve p_2 parametrelerinin geometrik anlamı:

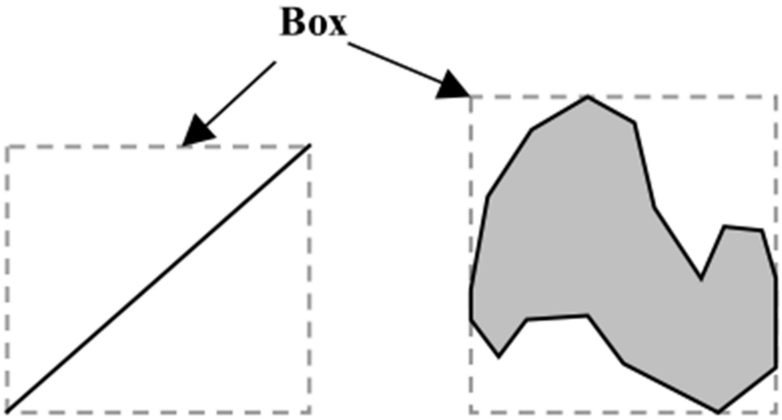
- $p_1 = 0$ ise, kesişim noktası i noktası ile çakışık
- $p_1 = 1$ ise, kesişim noktası j noktası ile çakışık
- $p_2 = 0$ ise, kesişim noktası k noktası ile çakışık
- $p_2 = 1$ ise, kesişim noktası l noktası ile çakışık

Yine programlama tekniği açısından p1 ve p2 parametrelerinin tam olarak sıfır ya da bir olması beklenemez. Bunun yanında sayısallaştırma hataları vb. gibi nedenlerle parametrelerin sıfırdan çok az küçük ya da birden çok az büyük olmaları durumunda kesişim noktasının doğru parçalarının ilgili uç noktaları ile çakışık olarak kabul edilmesi gerekebilir. Uygun olarak seçilen bir ε tolerans değerine göre parametrelerin irdelenmesi,

$$-\varepsilon \leq p_1 \leq 1 + \varepsilon \; ve - \varepsilon \leq p_2 \leq 1 + \varepsilon \quad (4.23)$$

biçiminde yapılmıştır. Ancak p_1 ve p_2 parametrelerinin birer orantı olup bir uzaklık ifade etmemelerinden dolayı ε tolerans değerinin seçimi çok kolay değildir. Bu noktada şöyle bir yaklaşım izlenmiştir. Tolerans değeri kabul edilebilir bir yuvarlatma hatası olarak alınmış, daha sonra koordinatları hesaplanan kesişim noktasının uç noktalarına olan uzaklıklarına bakılıp bu noktalarla çakışık kabul edilip edilemeyeceğine bakılmıştır.

Grafik programlama tekniğinde box testi olarak bilinen yöntemle, çizgisel ve alansal objelerin birbiri ile olan yakınlığı irdelenir. Kutu olarak da Türkçeleştirilebilecek box kavramı, çizgisel ya da alansal objenin dışına çizilen, objeyi kapsayan, koordinat eksenlerine paralel bir dikdörtgen objeyi ifade eder (Şekil 4.31). Box yardımıyla sadece mantıksal karşılaştırma yaparak iki çizgisel objenin kesişme olasılığı olup olmadığı, iki alansal objenin ise birbirini örtme olasılığının olup olmadığı belirlenebilir.



Şekil 4.31 Box kavramının grafik gösterimi

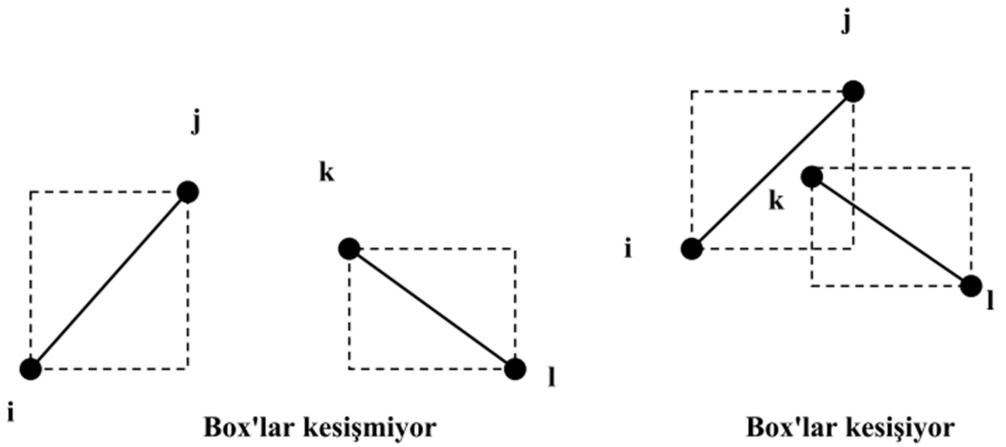
Box koordinatları ilgili objenin maksimum ve minimum koordinat değerlerinden elde edilebilir. i-j ve k-l doğru parçalarının kesişme olasılığını irdeleme durumunda box testi (Şekil 4.32),

$$\max(y_i, y_j) < \min(y_k, y_l) \tag{4.24}$$

ve

$$\max(x_i, x_j) < \min(x_k, x_l) \tag{4.25}$$

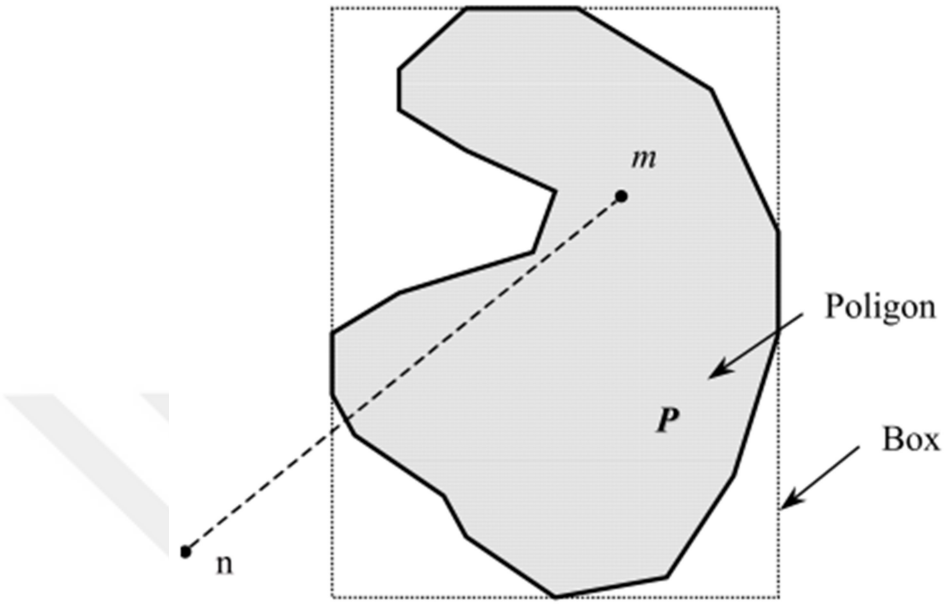
eşitliklerinin ikisi de sağlanıyorsa kesişme mümkün değil, sağlanmıyorsa kesişme mümkün, şeklinde sonuçlandırılmaktadır.



Şekil 4.32 Box testinin geometrik anlamı

Yapılan bu karşılaştırmalar sonucu kesişimle ilgili herhangi bir hesaplamaya gerek olmaması (Cromley, 1989; Bildirici, 2000), hesaplama sayısını ve hızını arttırmaktadır. Point in polygon testinde, bir noktanın bir alansal objenin (kapalı şeklin)

içinde olup olmadığını belirlemek için kapalı şeklin kesinlikle dışında olan bir yardımcı noktadan yararlanılır. Araştırılan nokta m, yardımcı nokta n olmak üzere m-n doğru parçasının, şekli kaç defa kestiği belirlenir (Şekil 4.33).



Şekil 4.33 Poligon içinde olup olmadığı araştırılan nokta (m), yardımcı nokta (n) ve box

Kesişim sayısı tek ise nokta şeklin içinde, çift ise dışındadır (Cromley, 1989). Point in polygon testi kendi içinde son derece basittir. Ancak arkasında programlama açısından çok da basit olmayan kesişim problemi vardır. Burada kesişim noktasının koordinatları ile değil yalnızca kesişimin var olup olmadığı ile ilgilenilmektedir. Kesişim sayısının araştırılmasından önce, m noktasının P poligonu içinde olup olamayacağının incelenmesi gerekir (Şekil 4.33). m noktası poligonu oluşturan box'ın içinde değilse poligonun da içinde olamaz. Box'ın sol alt köşe koordinatları poligonu oluşturan nokta koordinatlarının en küçükleri, sağ üst köşe koordinatları ise şekli oluşturan koordinatların en büyükleri olduğuna göre;

$$\min(x_i) > (x_m) > \max(x_i) \tag{4.26}$$

ve

$$\min(y_i) > (y_m) > \max(y_i) \tag{4.27}$$

şartlarının her ikisi de sağlanmıyorsa, nokta poligonun içinde olamaz. Şart sağlanıyorsa kesişim hesaplarına geçilebilir (Bildirici, 2000). Yardımcı n noktasının koordinatları, poligonu oluşturan box'dan yararlanarak şu şekilde hesaplanmıştır:

$$x_n = \min(x) - (\max(x) - \min(x)) \quad (4.28)$$

$$y_n = \min(y) - (\max(y) - \min(y)) \quad (4.29)$$

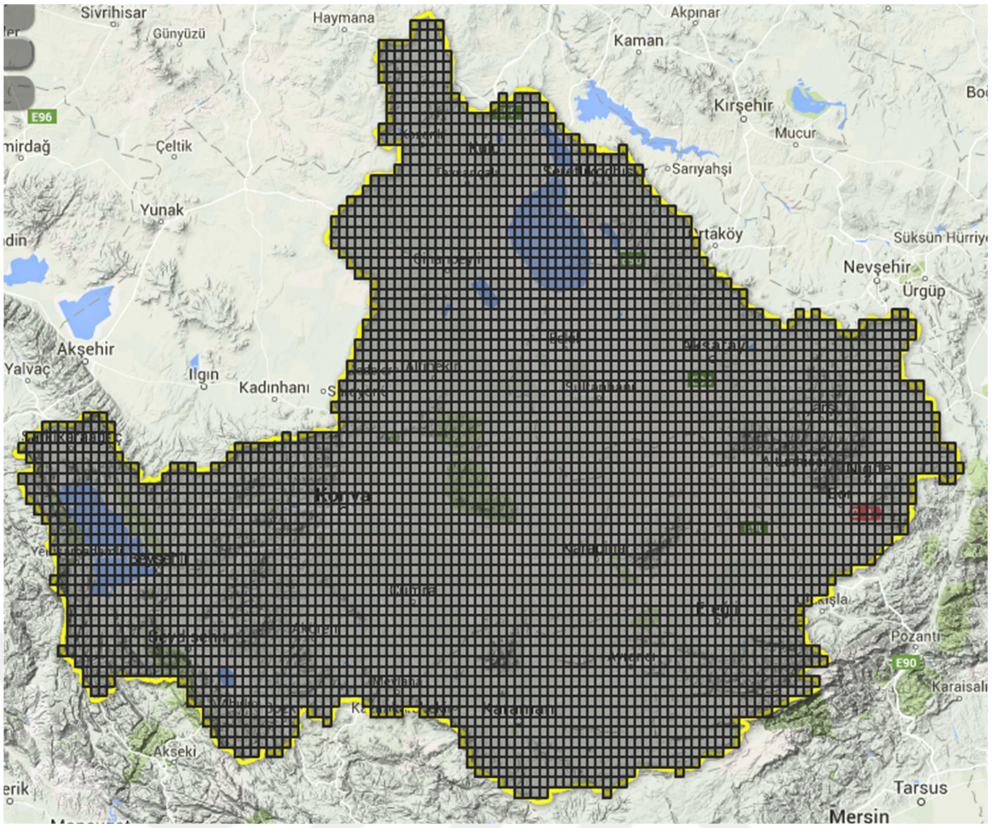
eşitliği ile n noktasının kesin olarak poligonun dışında olması sağlanmıştır. Uzakta bir n noktası seçmek yerine, eksenlerden birine paralel bir yardımcı doğru da seçilebilir (Cromley, 1989). Poligonu oluşturan kenar sayısı kadar kesişim hesabı yapılarak, kesişme sayısı belirlenmek zorundadır. Bu aşamada daha önce belirtildiği gibi kesişim öncesi box testi uygulanarak, m-n doğrusu ile ilgili kenarın kesişme olasılığı var ise hesaplanan d parametresi ile paralellik araştırılır. Paralel olma durumu yok ise p_1 ve p_2 parametreleri hesaplanır. Eğer,

$$0 > \text{p_1} > 1 \text{ ve } 0 > \text{p_2} > 1 \quad (4.30)$$

koşulu sağlanıyorsa, kesişim var, aksi halde yoktur. Kesişim noktasının koordinatlarına ihtiyaç olmadığından hesaplama bu noktada kesilir; kesişim sayısı bir artırılır ve bir sonraki kenara geçilir. Poligonu oluşturan tüm kenarlar için kesişim olup olmadığı araştırıldıktan sonra, toplam kesişim sayısı tek ise nokta poligonun içinde, çift ise değildir. Bu noktada bir üçüncü olasılık noktanın poligonu oluşturan kenarların birinin üzerinde olmasıdır. Nokta herhangi bir kenar üzerinde ise ya da nokta poligonu oluşturan noktalardan biri ile çakışık ise noktanın poligon dışında olduğu kararı verilmiştir.

4.3.6.2 İzoplet harita

Çizdirilmesine “Point in Polygon” testine göre karar verilen hücreler, GeoJSON ve KML formatlarında dosyalara yazılmıştır. Sonuçta oluşan 2’x2’ boyutlu hücrelerin görünümü Şekil 4.34’de verilmiştir. Oluşturulacak grid hücrelerinin boyutu, çıktı olarak üretilen dosya boyutunu doğrudan etkilemektedir. Dosya boyutu arttıkça grid katmanının harita ekranına eklenme süresi uzamakta, bunun yanında kullanıcıya katmanı her eklediğinde dosyayı tekrar indirmesi gibi bir yükle karşı karşıya gelmektedir. Tablo 4.4’de, grid dosyasının belirli hücre boyutlarına oluşan hücre sayısı ve dosya boyutu verilmektedir.



Şekil 4.34 Çalışma bölgesi için oluşturulan grid hücreleri

Daha önce de belirtildiği üzere, Google Maps JavaScript API kullanılarak harita üzerine eklenecek katmanların içeriği, katmanın yorumlanarak harita üzerinde gösterim süresini doğrudan etkilemektedir. Bu etki harita üzerine katman olarak eklenecek verinin yapısına da bağlıdır. Örneğin veri içerisinde bulunan obje sayısı ve bu objeyi oluşturan köşe sayısı arttıkça yorumlanması daha fazla zaman almaktadır.

Tablo 4.4 Hücre boyutuna göre oluşan nesne sayısı ve dosya boyutu

Kaynak Dosya			
KKH sınır	2040 nokta	45.7Kb	KML
Coğrafi Çözünürlük	**Hücre Sayısı**	**Dosya Boyutu**	**Dosya Formatı**
30” x 30”	73576 hücre	22.9MB	GeoJSON
1’ x 1’	18398 hücre	5.7MB	GeoJSON
1.5’ x 1.5’	8167 hücre	2.5MB	GeoJSON
2’ x 2’	4594 hücre	1.4MB	GeoJSON

Bu çalışmada kullanılacak olan dosya formatlarında (KML, GeoJSON) birbirine komşu iki hücre için ortak kenar iki defa çizilmektedir. Bu durumu ortadan kaldırmak için nesnelerin topolojilerini de dikkate alan bir JSON formatı olan TopoJSON

formatından yararlanılmıştır. Performanslarını test etmek amacıyla grid hücreleri her üç formatta da üretilerek oluşan dosya boyutları karşılaştırılmıştır. Dosyaların farklı formattaki boyutlarının karşılaştırması Tablo 4.5'de verilmiştir.

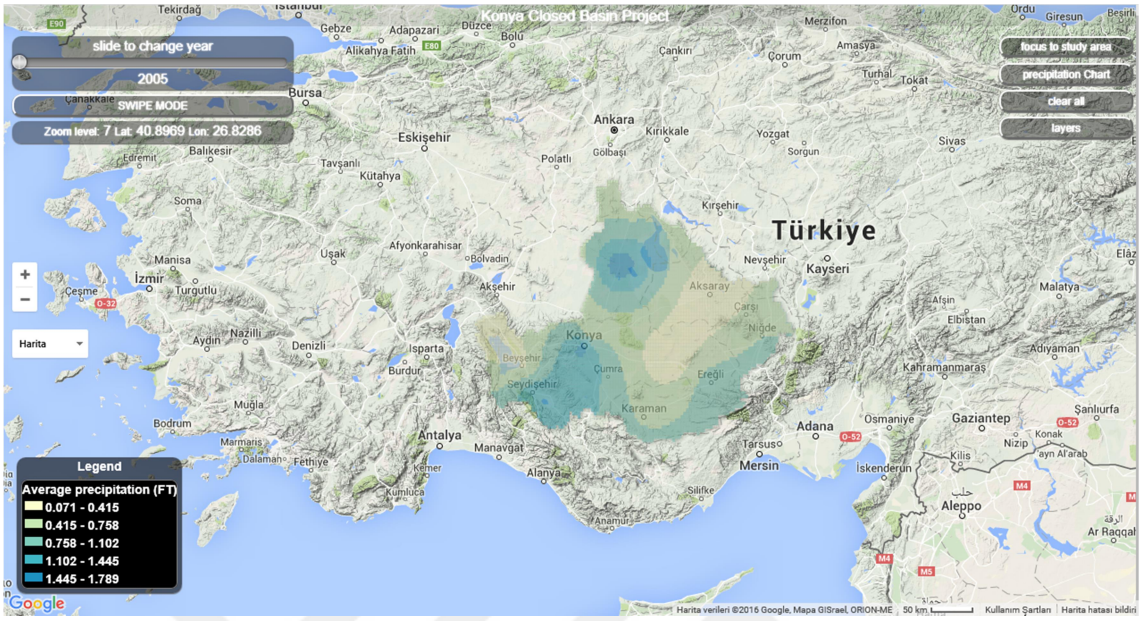
Tablo 4.5 Farklı formatta üretilen grid dosyası boyutları

Coğrafi Çözünürlük	**KML**	**GeoJSON**	**TopoJSON**
30"x30"	45.3MB	22.9MB	7.7MB
1'x1'	11.3MB	5.7MB	1.9MB
1.5'x1.5'	5MB	2.5MB	1.1MB
2'x2'	2.8MB	1.4MB	823Kb

Elde edilen sonuçlara göre TopoJSON formatının kullanılmasına karar verilmiştir. TopoJSON formatı KML ve GeoJSON dosya yapısındaki öznitelik bilgilerini depolama özelliğine de sahiptir. Ancak uygulamada grid hücrelerinin yeteri çözünürlükte kullanılabilmesi için TopoJSON formatının da yüksek boyutlu olduğuna karar verilmiştir. 30"x30" büyüklüğe sahip grid hücreleri için dahi 7.7MB dosya boyutu, sayfanın yüklenmesi ve yakınlaştırma gibi özelliklerin kullanılabilmesi için yetersiz kalmaktadır. Çözüm olarak oluşturulan grid hücrelerinin sunucu tarafında yorumlanması ve sonucun Google Maps altlık haritanın kullandığına benzer şekilde ekranda o an bulunan görüntü sınırlarına göre mozaikler halinde yüklenmesine karar verilmiştir. Bunu otomatik olarak gerçekleştiren Google Fusion Tables tabloları, verileri Google sunucularında tutup haritanın o an sahip olduğu yakınlaştırma düzeyine göre sunucu tarafında yorumlayıp kullanıcıya göndermesi oldukça kullanışlı bir özellik olarak tavsiye edilmektedir. Ancak Google Fusion Tables tabloları KML formatını desteklediği için oluşturulan grid dosyası KML formatında Fusion Tables tablolarına yüklenmiş ve haritaya katman olarak eklenmiştir. Bu işlem sonucunda da düşük yakınlaştırma düzeylerinde sunucu tarafında yorumlanan veride bazı eksiklikler meydana gelmektedir. Bunu gidermek için farklı çözünürlüklerde grid dosyaları hazırlanıp zoom seviyelerine göre bu farklı dosyaların yüklenmesi sağlanmıştır.

Grid dosyaları, veri tabanında merkez noktalarının coğrafi koordinatları da birer öznitelik değeri olacak şekilde oluşturulmuştur. Tematik katmanların oluşturulmasında Multiquadric enterpolasyon yöntemiyle (Hardy, 1992), harici bir dosyada tutulan kontrol noktalarından türetilen değerler, her bir hücrenin farklı bir öznitelik bilgisi olarak çıktı dosyalara eklenmiştir. Bu öznitelik bilgisinin tutulduğu alana dayalı

gerçekleştirilen biçimlendirme ile de renkli tematik katmanlar elde edilmiştir (Şekil 4.35).

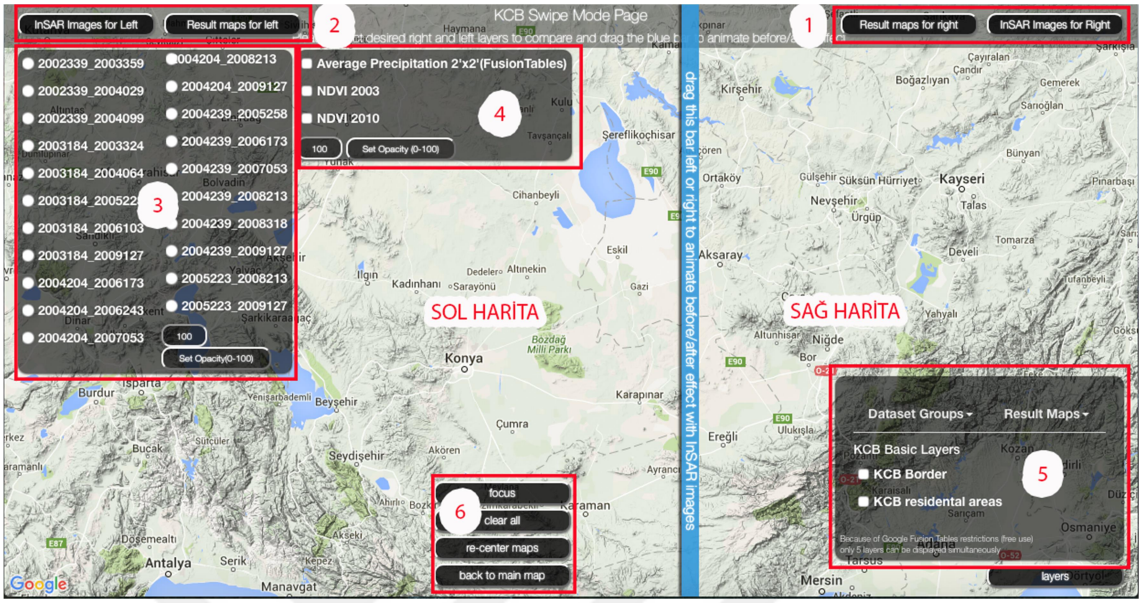


Şekil 4.35 Oluşturulan tematik katman

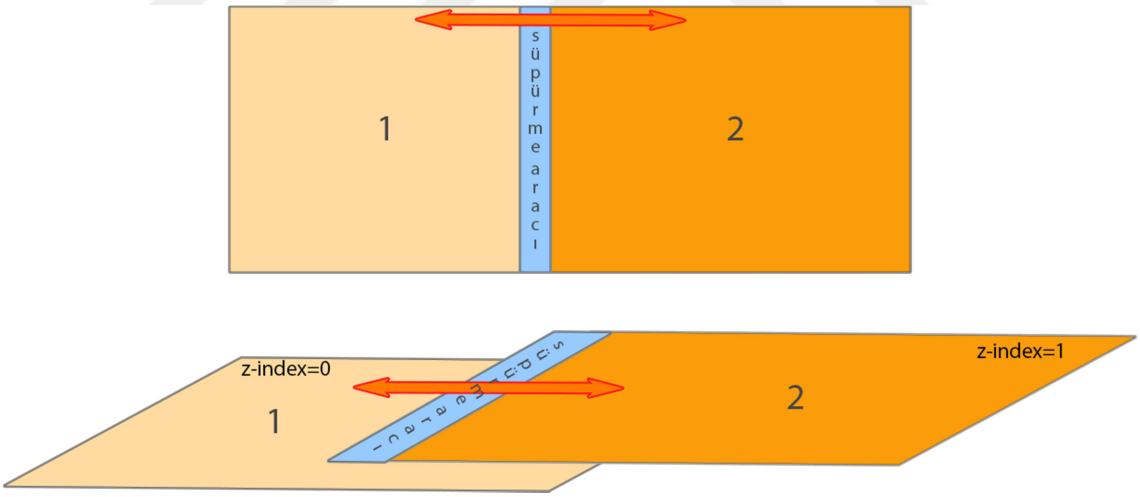
4.3.7 Katman karşılaştırma (Swipe Mode) ara yüzü

Tez çalışması kapsamında elde edilen sonuçların zamana bağlı karşılaştırılmasının yapılabilmesi amacıyla "Swipe Mode" adı verilen bir ara yüz tasarlanmıştır (Şekil 4.36). Kullanıcılar bu ara yüzü kullanarak, iki farklı katmanı tarihlerine bağlı olarak ara yüze ekleyebilmekte ve etkileşimli süpürme aracını kullanarak karşılaştırabilmektedir. Geliştirilen bu teknik katmalar arası değişimlerin izlenmesi açısından oldukça kullanışlıdır. Buradaki temel mantık birbiri üzerine bindirilen iki harita penceresinin aynı anda görüntülenmesidir (Şekil 4.37). Bu işlem, CSS dilinde yer alan z-index etiketi kullanılarak gerçekleştirilmektedir. z-index katman sıralama mantığına benzer şekilde web sayfası içerisindeki elemanların hangi sıra ile görüntüleneceğini belirler. Örneğin Şekil 4.37'da 1 numaralı div elemanının z-index değeri 0, 2 numaralı div elemanının z-index değeri 1 ya da daha büyük bir değeri temsil etmektedir. Ardından, elemanlardan biri sabit genişliğe (Şekil 4.37'de 1 numara ile gösterilen), diğeri de (Şekil 4.37'de 2 numara ile gösterilen) genişliğinin sürükle-bırak özelliği ile değiştirilebilir olacak şekilde kodlanması gerekmektedir. Süpürme efektinin oluşturulabilmesi için her iki harita da aynı noktaya merkezlenmeli ve yakınlaştırma derecesinin aynı değere ayarlanması gerekmektedir. Bu işlem kullanıcılarda sürekli bir

görüntüleme algısı oluşturarak aynı harita üzerinde veri işlem yapıldığı hissi uyandırmaktadır.



Şekil 4.36 Swipe mode arayüzü: ve araç menüleri (1) sağ harita için görüntü ekleme menüsü (2)sol harita için görüntü ekleme menüsü (3) tarihe göre görüntü çağırma (InSAR) (4)tarihe göre görüntü çağırma (NDVI) (5) katman ekleme menüsü ve (6) ana menü



Şekil 4.37 Swipe Mode ara yüzü çalışma mantığı

Swipe Mode ara yüzünde, ana harita sayfasına benzer iki harita penceresi bulunmaktadır. Buradaki fark kullanıcıların katman menüsünden çağırdığı katmanların her iki harita için de aynı anda yüklenmesi gerekliliğidir. Bu durum sadece ortak katmanlar için geçerlidir. Karşılaştırılması muhtemel katmanlar için sağ ve sol menüler oluşturulmuştur (Şekil 4.36 (1) ve (2)). Menüler aracılığıyla istenen tarihe ait analiz sonuçları sağ ve sol haritalar için ayrı ayrı yüklenebilmektedir (Şekil 4.36 (3) ve (4)).

Ayrıca iki haritanın tek bir harita gibi davranmasını sağlamak için kullanıcıların yakınlaştırma düzeyini değiştirmesi ve görüntü kaydırma işlemleri kısıtlanmıştır. Çünkü iki haritanın da eş zamanlı kaydırılması ya da yakınlaştırılması gerekmektedir. Bunun yerine, ara yüz içerisine eklenen katmanlara yakınlaşma fonksiyonunu çalıştıran düğmeler eklenmiştir. Ayrıca görüntü bindirme katmanları için saydamlık derecesinin ayarlanabileceği bir özellik de ilgili menülere eklenmiştir. Tüm bu araçlarla kullanıcılar analiz sonuçlarını, etkileşimli olarak saydamlık derecelerini belirleyerek, üzerine diğer katmanları ekleyebilerek, farklı tarihe ait sonuç görüntülerini karşılaştırabilme imkânına sahip olmaktadır.

5 ARAŞTIRMA BULGULARI VE TARTIŞMA

Temel amacı, mashup mimarisine dayanan ve CBS işlevine sahip bir mekânsal görsel analitik ortamının değerlendirilmesi, tasarlanması ve geliştirilmesi olan bu çalışmada önceki bölümlerde ele alınan hususlar, geliştirilen yöntem, teknik ve araçlar bu bölümde tartışılmaktadır. Öncelikle çalışmanın amacının yalnız KKH çalışma alanı kapsamında meydana gelen zemin hareketlerinin analizi olmadığını, gerçekleştirilen analizler ve mevcut verilerin yorumlanmasını, mekânsal görsel analitik bakış açısıyla, etkili ve etkileşimli görsel ara yüzlerle destekleyerek kolaylaştırmak olduğunun belirtilmesi gerekmektedir. Aynı zamanda, sonuçların çevrimiçi hizmetlerle web üzerinden dağıtımı da, tezin amaçları kapsamında ele alınan konuların başında gelmektedir. Tüm bu konular birlikte ele alındığında son on yılda geliştirilen teknolojilerle, literatürde oldukça fazla örneğine rastlanan harita mashup uygulamaları ön plana çıkmıştır. Öyle ki günümüzde bu gibi amaçlara ulaşmak için kullanılabilen masaüstü CBS paket programları için; genellikle hitap ettikleri kullanıcı profillerinin ticari ya da devlet kurumları ve büyük işletmeler olduğu düşünüldüğünde, pek de şaşırtıcı olmayan yüksek lisanslama bedelleri ile karşılaşılmaktadır. Diğer yandan bu gibi uygulamalar, kullanıcılarından sürekli geri beslemeler alarak ihtiyaçlar doğrultusunda gelişmiş araçlar tasarlamakta ve birçok işlevsellikleri bir arada sunmaktadır. Mashup uygulamalarında bu durum biraz farklıdır. Daha önce de belirtildiği gibi mashup uygulamaları uyarlamalı uygulamalar olup geliştirilen alternatif çözümler ortaya çıkan uygulamaların sunduklarıyla sınırlı kalmaktadır. Bunun sebebi, kullanılan servislerin sunulma amaçlarıdır. Örneğin bu çalışmada da kullanılan Google Maps API, gelişmiş bir web arama motorunun, kullanıcılarına konum tabanlı aramalar da yapabilecekleri bir servis oluşturulması mantığıyla ortaya çıkmıştır. Ardından Google, bu servisi kullanıcıların kendi sayfasına ekleyebilecekleri bir hizmet olarak geliştirmiş ve bu servisi kullanan farklı mashup uygulamaları internette yaygınlaşmaya başlamıştır.

Harita mashup uygulamaları aslında, farklı servisleri kümülatif olarak bir araya getirerek özel amaçlar için kullanılan harita uygulamalarına dönüştüren web geliştirme kavramıdır. Bu bakış açısıyla burada gerçekleştirilen çalışmada şu sorulara yanıt aranmıştır:

- Harita mashup uygulamalarının, web tabanlı CBS sistemleri gibi kullanılabilmesindeki potansiyel nedir?
- Bu uygulamalar için kullanılabilecek alternatif servisler nelerdir?
- Oluşturulacak uygulamada hangi yöntemler izlenmelidir?
- Servisler genel kullanıma ne ölçüde açıktır, açıksa ne gibi sınırlamalar mevcuttur.
- Çevrimiçi servisler kullanılırken özel verilerin gizliliği nasıl korunmaktadır?
- Web tabanlı harita uygulamalarında kullanılabilecek uygun mekânsal veri formatları nelerdir?
- Servisler tarafından sağlanmayan ek özellikler için ne gibi alternatif yöntemler geliştirilebilir?
- Tüm bu olanaklar kullanılarak kartografik gereksinimleri karşılayan mekânsal görsel analitik oluşturulmasında harita mashup uygulamaları literatürde nerede yer almalıdır?

5.1 Mevcut Harita Servislerinin Değerlendirilmesi

Çalışmada bu sorulara yanıt aranırken öncelikle oluşturulacak harita tabanlı mashup uygulamasında kullanılacak harita servisine odaklanılmıştır. Bu konuda farklı alternatifler mevcut olmakla birlikte, geliştirilecek uygulamanın amacı göz önünde bulundurularak tercih yapılabilmektedir. Bunlar genellikle iki sınıfta ele alınmaktadır. Birinci sınıfta ücretsiz ve açık kaynak kodlular ve ikinci sınıfta da ticari yazılım şirketleri tarafından oluşturulan, bazı kısıtlamalarla kullanıcılara ücretsiz olarak sunulan özel ara yüzler bulunmaktadır. Yapılan araştırmalarda birinci sınıfa örnek olarak OpenLayers, Leaflet, D3, Polymaps, ModestMaps, ikinci sınıfa örnek olarak da Google Maps, Yahoo! Maps, Bing Maps, MapQuest ve ArcGIS API ara yüzlerinin ön plana çıktığı belirlenmiştir. OpenLayers, web haritaları uygulamaları oluşturabilmek amacıyla kullanılabilen, yaygın ve zengin özelliklere sahip bir JavaScript API ara yüzüdür. Internet üzerinde oldukça fazla dokümantasyona ve örneğe sahiptir. Ancak bazı materyallerin yeni başlayan kullanıcılar için kavranması zor olabilmektedir. OpenLayers kullanmanın en önemli avantajlarından biri, bu API'yi kullanan oldukça büyük bir kitlenin bulunmasıdır. Bunun bir sonucu olarak Internet üzerinde kullanıcılar tarafından oluşturulmuş birçok forum ve örneğe ulaşabilmek mümkündür.

Leaflet, henüz yeni sayılabilecek bir web haritalama API ara yüzüdür ve yeni başlayanlar için kolay kavranma, mobil aygıtlarla iyi adapte olabilme ve hafif bir ara yüz olarak tasarlanmıştır. Dış kaynaklardan alınan mozaik haritalar ve istemci tarafı vektör grafik çizimlerine ağır bir önem vermektedir. Bu tabaka türlerini kullanan temel haritalar için Leaflet iyi bir seçim olmaktadır. Gerçekleştirilebilecek işlemlerin sınırlı olması nedeniyle çalışan örneklerinin sayısı çok fazla değildir. D3, genellikle grafik üretmek amacıyla kullanılan bir veri görselleştirme API ara yüzü olmasına rağmen, birçok harita örneği de içermektedir. Veri elemanlarını, Internet sayfasının belge nesne modeli ile bağlayarak (DOM), esnek veri animasyonları ve geçişlerini oluşturmaktadır. D3 etkileşimli haritalar ve grafikler ile bir web uygulaması oluşturmak için iyi bir seçenektir. Polymaps, öncelikle GeoJSON ve diğer kaynaklardan çizilmiş vektör nesneler ile yeni bir harita üretmek için tasarlanmış basit bir ücretsiz açık kaynak kodlu haritalama API ara yüzüdür. ModestMaps harita mozaikleri görüntülemek için oluşturulmuş küçük bir API ara yüzüdür. Açıklanan diğer API ara yüzleri ile karşılaştırıldığında, işlevsellik bakımından birçok eksiği bulunmaktadır. Yürütülen JavaScript kodları, bu kodların internet tarayıcısına aktarılmasına ihtiyaç duymaktadır. Sadece basit bir harita görüntüsü üretebilmek için dahi yüzlerce kodun bilinmesi gerekmektedir.

İkinci sınıf olarak kategorize edilen API ara yüzleri ticari yazılım şirketleri tarafından üretilenlerdir ve günümüzde çok yaygın olarak kullanılmaktadır. Konuma dayalı web tabanlı uygulamaların birçoğu bu tanınmış şirketlerin API ara yüzlerini kullanılarak üretilmektedir. Bu sınıfı diğerinden ayıran özellik API kaynak kodunun indirilememesi, değiştirilebilmesine izin verilmemesi ya da bir telif ödemeden dağıtımının yapılamamasıdır. Ancak önceki bölümde de açıklandığı üzere, bu API ara yüzlerinin neredeyse tamamı ticari bir amaç dışında kullanılması ya da yüksek miktarda internet trafiğine maruz kalmayan web tabanlı uygulamalarda kullanılması durumunda ücretsizdir. Ayrıca bu API ara yüzleri kullanılarak oluşturulan uygulamalarda, sağlayıcı şirket her zaman reklam ekleme seçeneğini saklı tutmaktadır. Bu sınıftaki en önemli örnekler Google Maps ve Microsoft Bing Maps olarak sıralanabilmektedir. Google Maps API, kullanıcılarına kendi ürettikleri veriyi Google tarafından sağlanan haritalar üzerine katman olarak ekleyebilme olanağı sunmaktadır. Eklenen veriler KML dosyaları aracılığı ile sağlanır ve kullanıcı tarafında oluşturulan etkileşimli vektör grafikleri olarak görüntülenir. Bu grafikler kullanıcı tarafından yeniden tasarlanabilmektedir ve “pop-up” pencereleri ya da tablolar eklenerek ek bilgiler

görüntülenebilmektedir. Google Maps API kullanımının en önemli avantajlarından biri de uygulamaya Google haritaları görünümü ve kullanım hissi veriyor olmasıdır. İnternet kullanıcılarının pek çoğu Google haritaları kullanım deneyimine sahiptir ve daha önce görmedikleri üçüncü parti bir uygulama içerisine yerleştirilmiş Google haritaları görünce daha rahat kullanım hissine sahip olacaktır. Google Maps API kullanımı açık kaynak kodlu diğer API ara yüzlerinden daha kolay ya da daha güçlü değildir, ancak tüm işlevsellikleri oldukça iyi belgelenmiştir ve oldukça büyük bir kullanıcı kitlesine sahiptir.

Microsoft Bing Maps bir diğer büyük ticari harita sağlayıcısıdır. Web ve mobil uygulamalar için API ara yüzleri mevcuttur ve bu ara yüzlerin hedefleri Google hedefleri ile örtüşmektedir. Google gibi Bing Maps de bir takım ticari kısıtlamalar hariç ücretsiz kullanım hakkı sunmaktadır. Google ve Bing Maps arasındaki farklardan biri Bing Maps uygulamalarının KML formatını daha az kullanıyor olmasıdır. Yahoo! Maps (şimdiki adıyla Here Maps) ve Mapquest şirketleri de Google ve Microsoft gibi harita verisini kullanıcılarının kendi web sayfalarına ekleyebilmelerini sağlayabilen API ara yüzleri sunmaktadır. Yahoo! Maps API üç farklı şekilde kullanılabilmektedir. Adobe Flash platformunu kullanan Flash API, Flash Plug-in kullanmadan etkileşimli haritaların üretilebileceği Ajax API ve XML formatını kullanan Simple API. MapQuest platformu da diğerlerine benzer şekilde, kullanıcılarına kendi Web sayfalarına Mapquest haritaları ekleyebilmeleri için API ara yüzüyle imkân sağlamaktadır. Mapquest Platformu'nun lisanslı ve açık veri olmak üzere iki sürümü vardır. Lisanslı sürümünde Mapquest Platformu, işletmeler ve geliştiriciler için, zengin mobil, web ve masaüstü haritalama çözümleri sunmak için geliştirici araçları, kaynaklar ve endüstri uzmanlığı sunmaktadır. MapQuest platformunun açık veri versiyonunun gücü öncelikle, tüm dünyanın düzenlenebilir haritasının oluşturulduğu OSM gelmektedir.

ESRI şirketi de ArcGIS platformuna uygun web haritalama API ara yüzleri geliştirmiştir. Bunlardan bazıları Google Maps API ve diğer ücretsiz açık kaynak kodlu API ara yüzleri ile karşılaştırıldığında, görece fonksiyon bakımından zengin olduğu söylenebilmektedir. API'ler, JavaScript, Flex, Silverlight, iOS ve Android gibi çoklu dilleri ve platformları desteklemektedir. Ürettiği birçok API arasında ArcGIS API for JavaScript tüm özelliklerle donatılanıdır. ArcGIS API'leri öncelikli olarak ArcGIS Online, Portal for ArcGIS ya da ArcGIS for Server kullanılarak yayınlanan Web servisleri ile birlikte kullanılmak için tasarlanmıştır. İncelenen tüm bu Web haritalama API ara yüzleri en çok kullanılanlarıdır ve güçlü haritalama araçları sağlamaktadır.

Ancak tüm gereksinimler göz önünde bulundurulduğunda hiçbiri Google Maps tarafından sağlanan ücretsiz içerik, zengin servis çeşitliliği ve ücretsiz API kadar etkili değildir. Çünkü Google birçok yan hizmeti haritalarıyla birleştirme imkânı sunarak zengin uygulamaların oluşturulmasına imkân sunmaktadır. Bu nedenle bu çalışmada Google altyapısını arkasına alan Google Maps API kullanılmıştır.

Bir diğer önemli konu da önceki bölümlerde detaylandırılan veri depolama ve yönetimidir. Bu işlem için Google altyapısında sunulan Fusion Tables servisi kullanılmıştır. Bu serviste veri Google sunucularında tutulmakta ve gerektiğinde bu sunuculardan yüklenmektedir. Bu servisin kullanılmasındaki amaç tablo yapısındaki veriyi kolayca tanıyabilmesi ve direkt olarak Google Maps API üzerine katman olarak eklenebilmesidir.

Veri görselleştirme işlemi için kullanılan diğer bir güçlü web-tabanlı araç da HighCharts API'dir. HighCharts, bir grafik üretme kütüphanesidir ve sadece JavaScript dilinde yazılmıştır. Highcharts line, spline, area, area spline, column, bar, pie, scatter, angular gauges, area range, area spline range, column range, bubble, box plot, error bars, funnel, waterfall and polar grafik türlerini desteklemektedir. Bu servis de diğerleri gibi ticari olmayan kullanımlar için ücretsizdir.

5.2 Sınırlamalar

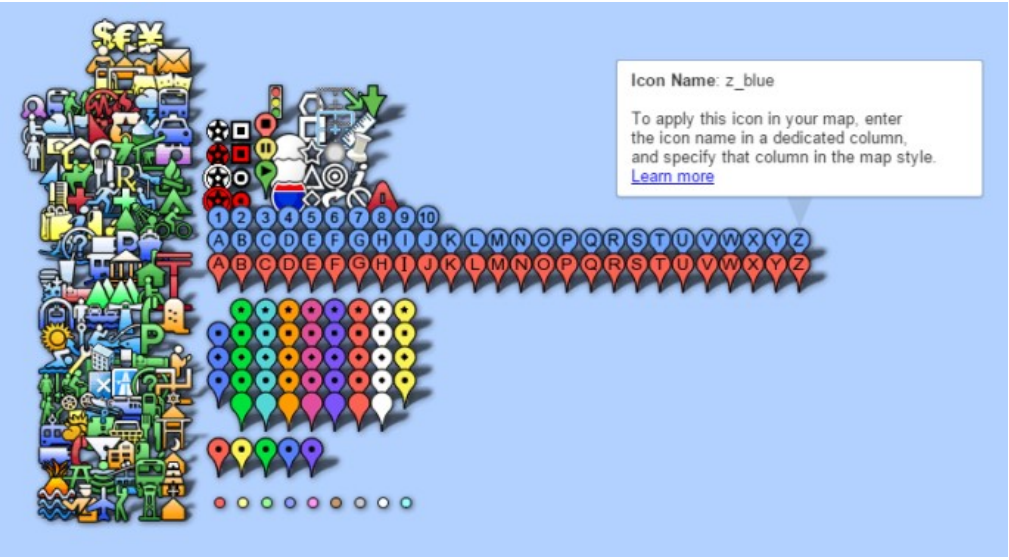
Yukarıda da bahsedildiği gibi ticari kullanımların önüne geçebilmek ve kurumsal çözümler pazarlayabilmek için birçok serviste olduğu gibi Google Maps JavaScript API servisinde de bazı kısıtlamalar[50] mevcuttur. Örneğin ardışık 90 gün içerisinde her 24 saatte 25.000 harita yüklemesi sınırlaması getirilmiştir. Birçok özel uygulama için bu sınırlama pek de önemli değildir. Aynı şekilde Fusion Tables servisi için de kullanım sınırlamaları mevcuttur. Bu kısıtlamalar;

- 25,000 sorgu/gün
- 250MB tablo başına alan
- Fusion Tables her bir hücresindeki veri maksimum 1 milyon karakteri desteklemektedir
- Tek bir http sorgusu için maksimum veri boyutu 1MB ile sınırlıdır
- Eklenecek tablo hücrelerinin toplam sayısı 10,000 hücredir

[50] https://developers.google.com/maps/documentation/javascript/usage

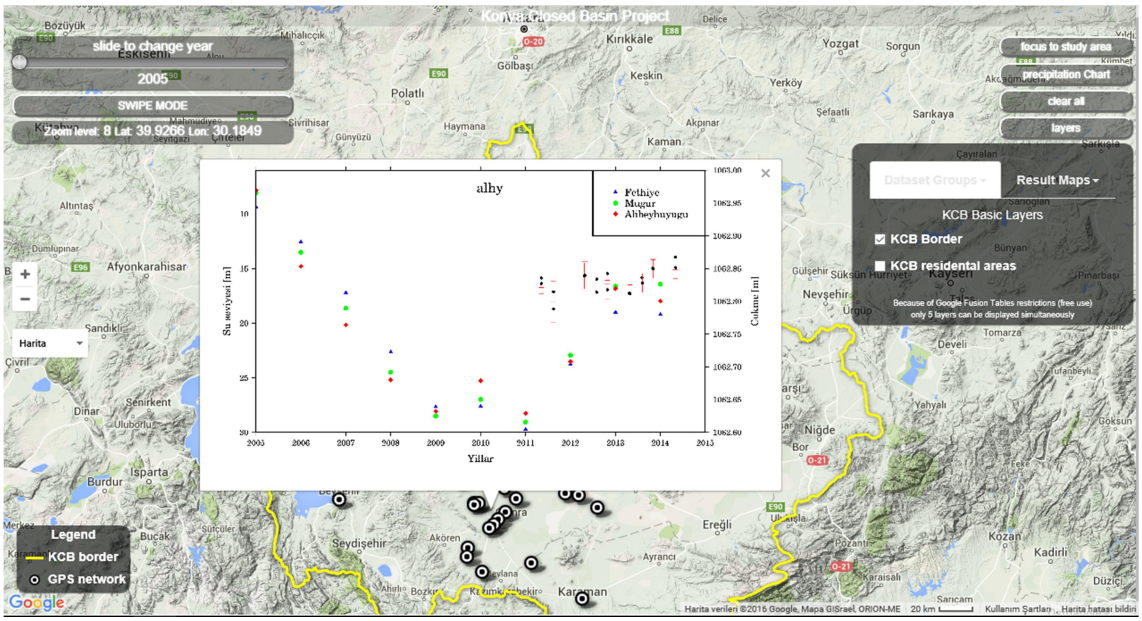
- Tek bir sorgu maksimum 500 INSERT ifadesi içerebilmektedir
- Tablodaki sadece ilk 100,000 kayıt haritalanabilmektedir ya da mekânsal sorgu sonuçlarında gösterilebilmektedir
- Her bir harita için aynı anda en fazla 5 katman eklenebilmektedir,

şeklindedir. Buradaki amaç da ticari kullanımları ücretlendirmektir. Ancak kısıtlamalar sadece veri depolama ve harita yükleme ile sınırlandırılmamalıdır. Kartografik açıdan da servisin sunduğu görselleştirme özellikleri ile sınırlamalar mevcuttur. Bunun nedeni de servis sunucularının, sunduğu hizmeti kullanıcılar için yeterli görmesidir. Örneğin noktasal nesneler için oluşturulan işaret ve renklendirmeler Fusion Tables içeriği ile sınırlıdır. Google, 200'ü aşkın nokta stili sunarak oldukça geniş bir stil kütüphanesine sahip olduğunu iddia etse de çoğu zaman bu işaretler kartografik tasarım açısından yetersiz kalmaktadır (Şekil 5.1). Ayrıca verilere ait öznitelik bilgilerinin bilgi pencereleriyle gösterim avantajına sahip Fusion Tables servisi bu işlem için bilgi kartları gibi otomatik çözümler sunmaktadır (Şekil 5.2). Bu kartlar özelleştirilebilir niteliktedir. Örneğin herhangi bir mekânsal nesnenin öznitelik bilgisi kartına bir web adresi girilerek çevrimiçi olarak sağlanan belgeler etkileşimli bilgi pencereleri ile kolaylıkla görüntülenebilmektedir. Ayrıca öznitelik bilgisi olarak çevrimiçi web adreslerine bağlantılar ve Google Charts servisi ile etkileşimli grafikler gibi farklı seçeneklere de ulaşmak mümkündür. Ancak bilgi pencerelerine tablo ya da sekmelerle ayrılmış sayfalar yerleştirmek gibi daha gelişmiş özelleştirmeler yapabilmek için HTML ve CSS kodlama bilgisine sahip olunması gerekmektedir.



Şekil 5.1 Google Fusion Tables standart nokta stilleri

Web servisleri kullanarak özel harita mashup uygulamalarında bir diğer sınırlama da elbette ki altlık olarak kullanılan haritanın kısıtlı özelleştirilme yetenekleridir. Genel anlamda kullanıcılar sadece harita gösterimi için kontrast, doygunluk ve renk seçeneklerini özelleştirme imkanına sahiptir[51]. Ancak bunun yanında birçok harita servisi uydu, yol ve arazi (topografik) gibi farklı harita türleri seçeneği de sunmaktadır.



Şekil 5.2 Fusion Tables servisi ile mekânsal bir nesnenin öznitelik bilgilerinin etkileşimli gösterimi.

Haritalarla kullanıcılar arasında iletişim köprüsü kuran lejantlar genellikle web haritalama servislerinde pek kullanılmamaktadır. Bunun nedeninin web haritalarının dinamik yapısı olduğu düşünülmektedir. Çünkü tüm web harita servisleri genellikle mozaik mantığı ile harita görüntülerini ya da vektörel katmanları yüklemekte, bu da her ölçekte ya da yakınlaştırma düzeyinde, her nesnenin görüntülenemeyeceği anlamına gelmektedir. Bu eksiklik neredeyse tüm web haritalama servislerinde mevcuttur. Bir takım çözümler mevcut olsa bile çok kullanışlı değildir. Örneğin Fusion Tables servisi ile oluşturulan tematik haritalar için otomatik lejant ekleme seçeneği bulunmaktadır. Bu özellik sadece tek bir konunun işlendiği haritalarda kullanılabilmektedir. Bu problem için geliştirilen yöntemle dinamik lejant oluşturulması mümkün kılınmıştır. Bu sayede harita üzerine eklenen mekânsal nesneler için lejant penceresinde açıklama eklenmekte, harita üzerinde bulunmayan nesneler için eklenmemektedir. Her ne kadar lejant

[51] https://developers.google.com/maps/documentation/javascript/styling

penceresi dinamik olsa da teknik açıdan yöntem statik bir yöntemdir. Bunun anlamı her bir mekânsal nesne için kullanılan işaretin, bir kütüphanede statik olarak tutulması, nesne haritaya eklendiğinde bu işaretin çağırılmasıdır. Haritaya yeni eklenen bir katman için kütüphaneye o katmanla ilgili işaretin eklenmesi ve katmanla ilişkilendirilmesi gerekmektedir.

Harita servisleri ayrıca, haritalar üzerine eklenecek mekânsal nesneler için bir katman mantığı sunmamaktadır. Varsayılan uygulamalarda API kodlamasıyla ilgili katmanlar harita üzerine eklenmekte ve harita web sayfasına yüklendiğinde birlikte gösterilmektedir. Ancak tüm bu geliştirmeler HTML, CSS ve JavaScript ortamlarında gerçekleştirildiği için farklı yaklaşımlar geliştirmek, ileri programlama düzeyleri gerektirebilmektedir. Bu problem için de tezde katman sınıflandırma yaklaşımı geliştirilmiş ve 4. bölümde detaylandırılmıştır.

5.3 Veri Gizliliği ve Kullanım Politikaları

Çevrimiçi hizmetlerle oluşturulan uygulamalarda veri gizliliği ve kulanım politikaları dikkat edilmesi gereken önemli bir konudur. Genellikle servislerin kullanım ve gizlilik politikaları, bu servisler ile yeni uygulamalar oluşturma aşamasında, kullanıcıyla servis sağlayıcısı arasında kabul edildiği varsayılan bir sözleşme yerine geçmektedir. Her servis sağlayıcısının kendine özgü kullanım şartları ve gizlilik politikaları mevcuttur. Ayrıca sunulan hizmetler, haritalar ve veriler bir takım telif haklarına tabi tutulabilmektedir. Bu nedenle servislerin kullanım koşullarının, gerçekleştirilecek uygulamanın gereksinimlerini karşılayıp karşılamadığı önceden belirlenmelidir. Çoğunluğu ticari amaçlı hizmet sunan bu servisler, kişisel bilgilerin hangi koşullarda başka amaçlar için kullanılabileceği ve üçüncü kişilerle paylaşabileceği konusunda detaylı bilgi sunmakla yükümlüdür.

Google, servislerinden herhangi birini ya da birkaçını kullanan kullanıcıların, Google tarafından yayınlanan kullanım şartları ve gizlilik politikasını[52] peşinen kabul etmiş ve bu politikaların, kullanıcı ile Google arasında bir sözleşme niteliği taşıdığını belirtmektedir. Buradaki amaç kullanıcıların Google servislerini izinsiz olarak yeniden dağıtması ve Google kaynaklı yeni bir ürün ya da hizmet oluşturmasını önlemektir. Kullanıcılar Google Maps servisini kullanırken Google tarafından şu lisanslara sahip olmaktadır:

[52] https://www.google.com/intl/tr/help/terms_maps.html

a. Haritaları görüntüleme ve açıklama ekleme;

b. KML dosyaları ve harita katmanları oluşturma;

c. Kamuya açık içeriği Google'a uygun atıfla çevrimiçi, video, veya basılı olarak yayınlama ve

d. Google Haritalar, Google Earth ve Street View kullanımına ilişkin izin sayfasında tanımlanan diğer birçok işlevi yerine getirme.

Ayrıca Google, kullanıcıları, Google haritaları üzerine eklenen içeriklerde ya da katmanlarda bulunabilecek fikri mülkiyet hakkı konusunda sorumlu tutmaktadır. Google, servislerinden herhangi birini kullanmak isteyen kullanıcılardan bir Google hesabı edinmelerini istemektedir. Bu hesapla kişisel verilerin, kullanıcı istemedikçe üçüncü kişilerle paylaşılmayacağını gizlilik politikasında[53] garanti etmektedir.

Oluşturulan mashup uygulamasında mekânsal verilerin depolanması ve yönetimi için Google Fusion Tables servisinin kullanılmasındaki bir neden de budur. Veriler Google Drive servisi aracılığı ile Fusion Tables sisteminde tutulmaktadır. Google Drive bulut-tabanlı bir sistem olup Google hesabına sahip herhangi bir kullanıcıya ücretsiz 15GB depolama alanı sunmaktadır. Daha önce de detaylandırıldığı üzere sayfa kaynağını görüntüleyen kullanıcılar sadece tablo ID numaralarını görmekte, veriye doğrudan erişememektedir. Bu sayede verilerin açık olarak dağıtılması ve üçüncü kişilerin eline geçmesi engellenmektedir. Google Drive servisi aynı zamanda kişisel verilerin yayınlanabilmesi için sunucu olarak da kullanılabilmektedir. Bu avantajla, oluşturulan kişisel harita uygulamalarının çevrimiçi olarak yayınlanabilmesi için harici bir sunucu kaynağı bulunması zorunluluğu da ortadan kalkmaktadır. HighCharts oldukça güçlü bir etkileşimli grafik servisi olmasına karşın depolama çözümleri sunmamaktadır. Aynı zamanda Fusion Tables API ile iletişim kuramamakta, bu servisten sağlanan veriyi görselleştirememektedir. Bu nedenle, bu servis kullanılırken veri HTML sayfası içerisinde tutulmakta ve tüm kullanıcıların görüntülemesine açık bulunmaktadır. Görselleştirilecek verilerin önem derecesine göre bu hususlar önceden belirlenmelidir.

5.4 Mashup Uygulamasının Değerlendirilmesi

Oluşturulan mashup uygulamasında, CBS benzeri fonksiyonelliklere sahip, web-tabanlı, tamamen ücretsiz servisleri kullanan ve kartografik gereksinimleri olabildiğince

[53] https://www.google.com/intl/tr/policies/privacy/index.html

karşılayabilecek bir yaklaşım sergilenmiştir. Aynı zamanda bu uygulamanın mekânsal görsel analitik araştırma alanının çekirdeğinde bulunan, büyük ve çok kaynaklı verilerin değerlendirilmesinde etkili ve etkileşimli görsel ara yüzlerin kullanılması felsefesini de karşılayacağı beklenmektedir. Ancak bu beklentilerin ne derece karşılandığının tartışılması gerekmektedir. Öncelikle web-tabanlı bir harita mashup uygulamasından, özellikle Google altyapısını kullanan bir uygulamadan, ne derece CBS fonksiyonellikleri beklenmesi gerektiğinin açıklanması gerekmektedir.

Günümüze kadar Google haritalarını herhangi bir amaç doğrultusunda en az bir kez bile kullanmamış insan sayısının oldukça az olduğu söylenebilmektedir. Çünkü en yakın eczanelerin bulunması, yakınlardaki petrol istasyonların gösterimi ya da mobil cihazlarla anlık konum bilgisi sorgulamaları gibi basit kullanımlar söz konusu olduğunda, başvurulan ilk kaynakların başında gelmektedir. Geleneksel masaüstü CBS yazılımlarıyla karşılaştırılması amacıyla ESRI tarafından geliştirilen yazılım paketlerine göz atmak gerekir. Çünkü bu işlemlerde en üst sıralarda yer alan çözümler geliştirmektedir. Her iki platformun da kullanıcı tercihlerine göre avantajları bulunmaktadır. Örneğin harici bir CBS sisteminin entegrasyonunda ESRI oldukça fazla seçenek sunarken Google bu konuda yetersiz kalmaktadır. İki platform da yüksek kaliteli altlık harita hizmeti sunmaktadır. Bazı bölgelerde biri diğerine üstünlük gösterse de genelde iki platform da bu işte gayet başarılı ürünlere sahiptir. ESRI son zamanlarda daha kullanıcı dostu ürünler geliştirme ve çevrimiçi hizmetler sunma gayretinde iken Google pek hızlı olmasa da CBS alanına doğru kararlı bir altyapı hazırlamaktadır. Ancak elbette ki konu mekânsal analizlere ya da verinin daha gelişmiş modellemesine geldiğinde ESRI ürünlerinin açık bir üstünlüğü söz konusudur. Bunun yanında, platformlar, birbiri ile rekabet içerisinde uygulamalara sahiptir. ArcGIS Explorer ve Google Earth, ArcGIS Online ve Google Maps Engine, ArcGIS web API ve Google Maps API bu uygulamalara örnektir. Her iki platform da şüphesiz bu alanda üst sıralardadır ve gelecekte daha gelişmiş çözümler sunacağına kesin gözüyle bakılmaktadır. ESRI ya da farklı bir CBS paket programında neredeyse tüm yaygın mekânsal veri formatı desteği bulunmaktadır. Bu da daha az format dönüşümü ve dolayısıyla veride daha az hata riski anlamına gelmektedir. Google birçok web harita servisinden daha fazla mekânsal veri formatı desteği sunmasına karşın CBS yazılımlarına kıyasla oldukça kısıtlı mekânsal analiz desteği sunmaktadır.

Harita mashup uygulamalarıyla bu gibi eksikliklere alternatif yöntemler geliştirilebilmektedir. Bu nedenle birinin diğerine üstünlüğü vardır denemez. Tamamen

kullanıcı ihtiyaçlarına göre karar verilmelidir. Tezde geliştirilen uygulamada da ihtiyaçlar doğrultusunda eksiklik görülen kısımlarda yeni yöntemler geliştirilmeye çalışılmıştır. Dinamik lejant, katman karşılaştırma, zaman kaydırma aracı ve izoplet katman yaklaşımı bu yöntemlere örnek olarak verilebilmektedir. Ancak daha önce de belirtildiği üzere geliştirilen araçlar uygulama odaklı olup pasif olarak geliştirilmiştir. Herhangi bir ara yüze sahip olmayan uygulamaların farklı uygulamalarda kolaylıkla kullanılabilmesi için kullanıcı ara yüzlerinin geliştirilmesi gerekmektedir.

Mekânsal görsel analitik bakış açısıyla, geliştirilen tüm bu araçların ve yaklaşımların birlikte kullanılması ile verinin ve analiz sonuçlarının yorumlanması için etkili ve etkileşimli görsel ara yüzlere sahip bir ortamın oluşturulduğu söylenebilmektedir. Ancak kartografik gereksinimlerin karşılanabilmesi açısından çok da esnek davranılamamıştır. Çünkü harita tasarımı konusunda, kullanılan servise bağlı kısıtlamalar oldukça fazladır ve az sayıda özelleştirme seçeneği sunulmaktadır. Mekânsal analizler ve gelişmiş veri modelleme konusunda Google Maps ve diğer servisler basit sorgulamalar dışında fazla seçenek sunamamaktadır. Bu nedenle zemin çökmeleri, NDVI haritaları ve InSAR görüntü çiftleri gibi mekânsal analiz işlemleri üçüncü parti uygulamalarla gerçekleştirilip sonuçlar raster formatında sisteme dâhil edilmiştir. Elbette bu gibi işlemler sonucu harita üzerine eklenen katmanlar statik görüntülerden oluşmaktadır ver zoom seviyesine göre dinamik olarak ölçeklenme ya da bilgi pencereleri gibi etkileşimli özellikler eklenememektedir. Ayrıca sunulan tematik harita olanakları oransal sembol ve koroplet haritalar gibi kısıtlı seçenekleri içermektedir. Kontrol noktalarından ara değerlerin türetildiği ve bu değerlere göre renklendirilerek görselleştirilen izoplet ya da izaritmik haritalar gibi seçenekler birçok haritalama servisinde sunulmamaktadır.

Bu tezde geliştirilen izoplet katman yaklaşımı, bu probleme çözüm getirmek amacıyla sunulmuştur. Neredeyse tüm web tabanlı haritalama servislerinin desteklediği bir formatta (JSON) çıktı üretebilen algoritma sayesinde, sınırları belirli bir alanda dağılmış kontrol noktalarından ara değerler hesaplanmakta ve detayları 4. bölümde açıklandığı üzere alanı kullanıcının belirlediği sıklıkta eşit alanlara bölen grid hücrelerine bu ara değerleri atamaktadır. Ardından HTML ve CSS özellikleri kullanılarak bu hücreler renklendirilmekte ve tematik katmanlar oluşturulmaktadır. Bu yöntemle oluşturulan katmanlar vektörel özellikte olup dinamik ölçekleme ve bilgi pencereleri gibi etkileşimli işlemlere izin vermektedir. Tez çalışmasının ön önemli

katkılarından biri sayılabilecek bu çözüm sayesinde istatistiksel yüzeyler oluşturulup, çevrimiçi haritalar üzerine katman olarak eklenebilmektedir.

Yukarıda sayılan tüm olumsuzluklara rağmen, geliştirilen yöntem ve araçlarla çalışmada hedeflenen amaçlara ulaşılmıştır.

6 SONUÇLAR VE ÖNERİLER

Web 3.0 versiyonunun semantik tabanlı olacağı söylentilerinin gittikçe arttığı bu günlerde, gelecekte çok güçlü harita uygulamalarıyla karşılaşılacağı sürpriz olmayacaktır. Günümüz teknolojilerinin ve olanaklarının ele alındığı bu tezde *"mashup ekosistemi"* (Weiss ve Gangadharan, 2010) için farklı yaklaşımlar geliştirilmeye çalışılmıştır. Tezde, mashup yaklaşımına dayanan ve CBS fonksiyonelliklerine sahip bir mekânsal görsel analitik ortamının değerlendirilmesi, tasarlanması ve geliştirilmesi amaçlanmıştır. KKH sınırları içerisinde farklı kaynaklardan elde edilen verilerle gerçekleştirilen analizlerin sonuçlarının, web üzerinden dağıtımının yapılabilmesi için Google Map API V3, Google Fusion Tables API, AJAX, Google Drive ve High Charts API servisleri kullanılmıştır. Uygulamada aynı zamanda esneklik ve daha fazla etkileşim için HTML, CSS ve JavaScript gibi web istemci teknolojileri de kullanılmıştır.

Bu teknolojiler kullanılarak web-tabanlı harita servislerinin hali hazırda sunmadığı, dış verinin katman olarak eklenmesinde katman gruplandırma ve eklenen katmanların dinamik lejant alanlarının oluşturulması problemine, HTML, CSS ve JavaScript özellikleri kullanılarak bir çözüm geliştirilmiştir. Haritaya eklenecek katmanlar gruplandırılarak her bir grup için **`div`** elamanları oluşturulmuştur. Bu elemanların davranışları JavaScript kodları kullanılarak programlanmıştır. Dinamik lejant yaklaşımı aynı zamanda harita alanının sınırlandırılmasının da önüne geçmektedir. Buradaki dezavantaj her bir katman için belirlenen özel işaret imajlarının sabit oluşudur. Herhangi bir katmanın işareti değiştirildiğinde lejant penceresindeki karşılığı otomatik olarak yenilenememektedir. Ancak bu durum kullanıcıların harita işaretlerine müdahale edememesi açısından herhangi bir sorun teşkil etmemektedir ancak farklı kullanıcı gruplarını hedef alan farklı amaçlı uygulamalar için daha gelişmiş yöntemler geliştirmek gerekmektedir.

Ayrıca genel amaçları konum paylaşmak, konum tabanlı aramalar ve reklam yayınları olan web harita servisleri ile gelişmiş mashup uygulamaları oluşturulması konusunda, mevcut servislerin henüz sunmadığı izoplet veri katmanlarının oluşturulması problemine yeni bir yaklaşımla çözüm sunulmaya çalışılmıştır. "Point in Polygon" ve "Multikuadrik Radyal Bazlı Fonksiyon" yaklaşımının kullanıldığı çözüm ile kullanıcılar, istatistiksel yüzeyleri oluşturabilmekte ve web-tabanlı harita servislerine

katman olarak ekleyerek izoplet haritalar oluşturabilmektedir. Harita üzerine eklenen katmanların zamana bağlı olarak değişimlerinin izlenmesi için "Swipe Mode" adı verilen yeni bir ara yüz tasarlanmıştır. Bu ara yüz ile iki farklı katman eş zamanlı olarak harita üzerine eklenebilmekte ve geliştirilen ekran süpürme aracıyla katmanlar arasındaki değişimlerin izlenmesi sağlanmaktadır. Bu özelliğin kullanılabilmesi için birbiri üzerine örtüşen iki farklı harita penceresi kullanılmıştır. Haritalardan biri sabit diğeri ise bir kaydırma çubuğu ile genişliğinin değiştirilebilir olması sağlanmıştır. Her iki harita da aynı noktaya, aynı zoom seviyesi ile merkezlendiğinde kullanıcıya tek bir harita üzerinde iki farklı katman görüntülüyor hissi verilmeye çalışılmıştır.

Zamana bağlı değişimlerin izlenmesi için geliştirilen bir diğer yaklaşım ise zaman kaydırma çubuğu aracıdır. Bu araç, harita üzerine eklenecek katmanların zamana bağlı olarak çağırılmasını sağlamaktadır. Bunun için JavaScript ve JQuery özelliklerinden faydalanılmıştır. Bunun için HTML ve CSS özellikleri kullanılarak oluşturulan bir kaydırma çubuğuna belirli bir aralıkta değer kümeleri atanmıştır. Kullanıcı bu çubuk üzerindeki düğmeyi kaydırarak farklı değer aralığına ulaştığında, bir sorgu oluşturulmakta ve bu sorgu ile farklı Fusion Tables tabloları harita üzerine eklenebilmektedir. Daha önce de belirtildiği üzere oluşturulan tüm görselleştirme bileşenleri herhangi bir web tarayıcısında çalışabilmektedir. Çalışmada geliştirilen yaklaşımların farklı kullanıcılar tarafından kullanılabilecek formata getirilebilmesi için, gelecekte ayrı ara yüzlere sahip araçlar olarak tasarlanması ve web üzerinden dağıtımının yapılması planlanmaktadır. Örneğin bu sayede kullanıcılar kendi çalışma bölgelerini ara yüzden seçeceği hücre boyutuna göre grid hücrelerine ayırarak yine aynı ara yüzden belirleyeceği renk paletine göre tasarım yapılarak tematik katmanlar elde edebilecektir. Ayrıca izoplet katman yaklaşımına farklı enterpolasyon yöntemleri eklenerek aracın geliştirilmesi de düşünülmektedir. Bu yaklaşımı kullanarak kullanıcılar sürükle-bırak yöntemi ile kendi verilerini karşılaştırabilecektir. Aynı zamanda dinamik lejant gösteriminin de haritaya eklenen içerikten otomatik olarak oluşturulması için geliştirilmesi planlanmaktadır. Bu çalışmada, imkânlar dâhilinde kısıtlı çözümler sunulan zamana bağlı görselleştirme, gelecekte daha detaylı ele alınması gereken önemli bir konudur. Tez çalışmasının bir bütün olarak ele alındığında, gelecekte harita mashup uygulamaları alanında çalışacak araştırmacılara önemli yönlendirmeler sunulduğu düşünülmektedir.

KAYNAKLAR

Aitchison, A., 2011, The Google Maps/Bing Maps Spherical Mercator Projection, *Available onine: http://alastairawordpresscom/2011/01/23/the-google-maps-bing-maps-spherical-mercator-projection/(accessed on 14 April 2014)*.

Al-Ahmadi, K., Al-Zahrani, A. ve Al-Dossari, A., 2013, A Web-Based Cancer Atlas of Saudi Arabia, *Journal of Geographic Information System*, 05 (05), 471-485.

Albuquerque, J. P., Herfort, B., Brenning, A. ve Zipf, A., 2015, A geographic approach for combining social media and authoritative data towards identifying useful information for disaster management, *International Journal of Geographical Information Science*, 29 (4), 667-689.

Andrienko, G., Andrienko, N., Jankowski, P., Keim, D., Kraak, M. J., Maceachren, A. ve Wrobel, S., 2007, Geovisual analytics for spatial decision support: Setting the research agenda, *International Journal of Geographical Information Science*, 21 (8), 839-857.

Andrienko, N., Andrienko, G. ve Gatalsky, P., 2003, Exploratory spatio-temporal visualization: an analytical review, *Journal of Visual Languages and Computing*, 14 (6), 503-541.

Andrienko, N. ve Andrienko, G., 2012, A visual analytics framework for spatio-temporal analysis and modelling, *Data Mining and Knowledge Discovery*.

Battersby, S. E., Finn, M. P., Usery, E. L. ve Yamamoto, K. H., 2014, Implications of web Mercator and its use in online mapping, *Cartographica: The International Journal for Geographic Information and Geovisualization*, 49 (2), 85-101.

Batty, M., Hudson-Smith, A., Milton, R. ve Crooks, A., 2010, Map mashups, Web 2.0 and the GIS revolution, *Annals of GIS*, 16 (1), 1-13.

Bearman, N. ve Appleton, K., 2012, Using Google Maps to collect spatial responses in a survey environment, *Area*, 44 (2), 160-169.

Bedrina, T., Parodi, A., Quarati, A. ve Clematis, A., 2012, ICT approaches to integrating institutional and non-institutional data services for better understanding of hydro-meteorological phenomena, *Natural Hazards and Earth System Sciences*, 12 (6), 1961-1968.

Benke, K. K., Sheth, F., Betteridge, K., Pettit, C. J. ve Aurambout, J. P., 2014, Application of geovisual analytics to modelling the movements of ruminants in the rural landscape using satellite tracking data, *International Journal of Digital Earth*, 8 (7), 579-593.

Benslimane, D., Dustdar, S. ve Sheth, A., 2008, Services mashups: The new generation of web applications, *IEEE Internet Computing*, 12 (5), 13.

Bildirici, I. O., 2015, The Web Mercator Projection: A Cartographic Analysis, *1st ICA European Symposium on Cartography*, Vienna, Austria, 221-231.

Bildirici, İ. Ö., 2000, 1: 1000-1: 25 000 Ölçek Aralığında Bina Ve Yol Objelerinin Sayısal Ortamda Kartografik Genelleştirmesi, *Fen Bilimleri Enstitüsü*.

Boni, R., Herrera, G., Meisina, C., Notti, D., Bejar-Pizarro, M., Zucca, F., Gonzalez, P. J., Palano, M., Tomas, R., Fernandez, J., Fernandez-Merodo, J. A., Mulas, J., Aragon, R., Guardiola-Albert, C. ve Mora, O., 2015, Twenty-year advanced DInSAR analysis of severe land subsidence: The Alto Guadalentin Basin (Spain) case study, *Engineering Geology*, 198, 40-52.

Castelletto, N., Gambolati, G. ve Teatini, P., 2013, Geological CO2 sequestration in multi-compartment reservoirs: Geomechanical challenges, *Journal of Geophysical Research-Solid Earth*, 118 (5), 2417-2428.

Chow, T. E., 2008, The potential of maps APIs for internet GIS applications, *Transactions in GIS*, 12 (2), 179-191.

Chow, T. E., 2011, Geography 2.0: A mashup perspective, *Advances in web-based GIS, mapping services and applications*, 15-36.

Cook, K. A. ve Thomas, J. J., 2005, Illuminating the path: The research and development agenda for visual analytics, *Pacific Northwest National Laboratory (PNNL), Richland, WA (US)*.

Cormode, G. ve Krishnamurthy, B., 2008, Key differences between Web 1.0 and Web 2.0, *First Monday*, 13 (6).

Cromley, R., 1989, Digital cartography, Prentice-Hall, p.

Dangermond, J., 2002, Web services and GIS, *Geospatial Solutions*, 12 (7), 56-57.

Daniel, F., Soi, S., Tranquillini, S., Casati, F., Heng, C. ve Yan, L., 2012, Distributed orchestration of user interfaces, *Information Systems*, 37 (6), 539-556.

Drigas, A., Koukianakis, L. ve Papagerasimou, Y., 2011, Towards an ICT-based psychology: E-psychology, *Computers in Human Behavior*, 27 (4), 1416-1423.

Du, Z. Y., Ge, L. L., Li, X. J. ve Ng, A. H. M., 2016, Subsidence monitoring in the Ordos basin using integrated SAR differential and time-series interferometry techniques, *Remote Sensing Letters*, 7 (2), 180-189.

DuVander, A., 2010, Map Scripting 101: An example-driven guide to building interactive maps with Bing, Yahoo!, and Google Maps, No Starch Press, p.

Eick, S. G., 2010, Geospatial visualization with VisTracks, *Wiley Interdisciplinary Reviews: Computational Statistics*, 2 (3), 272-286.

Field, K. ve O'Brien, J., 2010, Cartoblography: Experiments in Using and Organising the Spatial Context of Micro-blogging, *Transactions in GIS*, 14, 5-23.

Flanagin, A. J. ve Metzger, M. J., 2008, The credibility of volunteered geographic information, *GeoJournal*, 72 (3-4), 137-148.

Garreau, L., Mouricou, P. ve Grimand, A., 2015, Drawing on the Map: An Exploration of Strategic Sensemaking/Giving Practices using Visual Representations, *British Journal of Management*, 26 (4), 689-712.

Garrett, J. J., 2005, Ajax: A new approach to web applications.

Ge, L. L., Ng, A. H. M., Li, X. J., Abidin, H. Z. ve Gumilar, I., 2014, Land subsidence characteristics of Bandung Basin as revealed by ENVISAT ASAR and ALOS PALSAR interferometry, *Remote Sensing of Environment*, 154, 46-60.

GeoRSS, 2016, Geographically Encoded Objects for RSS feeds, http://www.georss.org/: [02.01.2016].

Ghani, S., Kwon, B. C., Lee, S., Yi, J. S. ve Elmqvist, N., 2013, Visual analytics for multimodal social network analysis: a design study with social scientists, *IEEE Trans Vis Comput Graph*, 19 (12), 2032-2041.

Github-contributors, 2016, TopoJSON: An extension to GeoJSON that encodes topology. , https://github.com/mbostock/topojson: [22.01.2016].

Gonzalez, H., Halevy, A., Jensen, C. S., Langen, A., Madhavan, J., Shapley, R. ve Shen, W., 2010, Google fusion tables: data management, integration and collaboration in the cloud, *Proceedings of the 1st ACM symposium on Cloud computing*, 175-180.

GoogleDevelopers, 2016a, Which API Do I Need?, https://developers.google.com/maps/documentation/api-picker: [11.01.2016].

GoogleDevelopers, 2016b, Google Maps JavaScript API, https://developers.google.com/maps/documentation/javascript/maptypes: [21.02.2016].

GoogleDevelopers, 2016c, Keyhole Markup Language, https://developers.google.com/kml/documentation/kml_tut: [09.02.2016].

Han, H., Xue, Y. X. ve Oyama, K., 2013, Mashup Technology: Beyond Open Programming Interfaces, *Computer*, 46 (12), 96-99.

Hardy, R. L., 1992, Comments on the Discovery and Evolution of the Multiquadric Method, *Computers & Mathematics with Applications*, 24 (12), R11-R12.

Howard Butler, M. D., Allan Doyle, Sean Gillies, Tim Schaub, Christopher Schmidt 2016, The GeoJSON Format Specification, http://geojson.org/geojson-spec.html: [09.02.2016].

Huang, C. Y. ve Liang, S. H. L., 2013, LOST-Tree: a spatio-temporal structure for efficient sensor data loading in a sensor web browser, *International Journal of Geographical Information Science*, 27 (6), 1190-1209.

International Cartographic Association, I. C. A., 2003, A Strategic Plan for the International Cartographic Association, ICA 2003-2011, http://icaci.org/files/documents/reference_docs/ICA_Strategic_Plan_2003-2011.pdf: [07.02.2016].

International Cartographic Association, I. C. A., 2011, Strategic Plan for the International Cartographic Association, ICA 2011-2019, http://icaci.org/files/documents/reference_docs/ICA_Strategic_Plan_2011-2019.pdf: [07.02.2016].

Ishitsuka, K., Fukushima, Y., Tsuji, T., Yamada, Y., Matsuoka, T. ve Giao, P. H., 2014, Natural surface rebound of the Bangkok plain and aquifer characterization by persistent scatterer interferometry, *Geochemistry Geophysics Geosystems*, 15 (4), 965-974.

Karnatak, H. C., Shukla, R., Sharma, V. K., Murthy, Y. V. S. ve Bhanumurthy, V., 2012, Spatial mashup technology and real time data integration in geo-web application using open source GIS - a case study for disaster management, *Geocarto International*, 27 (6), 499-514.

Keim, D., Andrienko, G., Fekete, J. D., Gorg, C., Kohlhammer, J. ve Melancon, G., 2008, Visual analytics: Definition, process, and challenges, *Information Visualization: Human-Centered Issues and Perspectives*, 4950, 154-175.

Keim, D. A., Mansmann, F., Schneidewind, J. ve Ziegler, H., 2006, Challenges in visual data analysis, *Information Visualization, 2006. IV 2006. Tenth International Conference on*, 9-16.

Kono, N., Arakawa, K., Ogawa, R., Kido, N., Oshita, K., Ikegami, K., Tamaki, S. ve Tomita, M., 2009, Pathway projector: web-based zoomable pathway browser using KEGG atlas and Google Maps API, *Plos One*, 4 (11), e7710.

Kraak, J.-M., 2015, Keynote by Menno-Jan Kraak, Euro Carto 2015, Viyana, AVUSTURYA, http://eurocarto.org/keynote/: [05.03.2016].

Kraak, M.-J., 2001, Settings and needs for web cartography, In: Web cartography: Developments and prospects, Eds: Kraak, M.-J. B., Allan: Taylor & Francis Group, London, p. 1-7.

Kveladze, I., Kraak, M. J. ve Van Elzakker, C. P. J. M., 2015, The space-time cube as part of a GeoVisual analytics environment to support the understanding of movement data, *International Journal of Geographical Information Science*, 29 (11), 2001-2016.

Lee, N. ve Rojas, E. M., 2013, Visual representations for monitoring project performance: Developing Novel prototypes for improved communication, *Journal of Construction Engineering and Management*, 139 (8), 994-1005.

Leopold, C., 2015, Visual Representations for Spatial Thinking, In: The Visual Language of Technique, Eds: Springer, p. 37-56.

Li, S., 2008, Web mapping/GIS services and applications, In: Advances in photogrammetry, remote sensing and spatial information sciences, Eds: Zhilin Li, J. C., Emmanuel Baltsavias: Taylor & Francis Group, London, p. 335-353.

Li, S., Veenendaal, B. ve Dragićević, S., 2011, Advances, challenges and future directions in web-based GIS mapping services, *Li, S, Dragićević, S and Veenendaal B (2011) Advances in Web-based GIS, Mapping Services and Applications, Taylor & Francis Group, London, ISBN*, 978-970.

Liu, G. X., Li, J., Xu, Z., Wu, J. C., Chen, Q. A., Zhang, H. X., Zhang, R., Jia, H. G. ve Luo, X. J., 2010, Surface deformation associated with the 2008 Ms8.0 Wenchuan earthquake from ALOS L-band SAR interferometry, *International Journal of Applied Earth Observation and Geoinformation*, 12 (6), 496-505.

Liu, S. B. ve Palen, L., 2010, The New Cartographers: Crisis Map Mashups and the Emergence of Neogeographic Practice, *Cartography and Geographic Information Science*, 37 (1), 69-90.

Liu, S. Y., Pu, J. S., Luo, Q., Qu, H. M., Ni, L. M. ve Krishnan, R., 2013, VAIT: A Visual Analytics System for Metropolitan Transportation, *Ieee Transactions on Intelligent Transportation Systems*, 14 (4), 1586-1596.

Llorà, X., Sastry, K., Alías, F., Goldberg, D. E. ve Welge, M., 2006, Analyzing active interactive genetic algorithms using visual analytics, *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, 1417-1418.

Luo, W. ve MacEachren, A. M., 2014, Geo-social visual analytics, *Journal of Spatial Information Science* (8).

Macdonald, S., 2008, Data Visualisation Tools: Part 2-Spatial Data in a Web 2.0 Environment and Beyond.

MacEachren, A. M., Gahegan, M., Pike, W., Brewer, I., Cai, G., Lengerich, E. ve Hardistry, F., 2004, Geovisualization for knowledge construction and decision support, *Computer Graphics and Applications, IEEE*, 24 (1), 13-17.

MacEachren, A. M., 2013, Cartography as an Academic Field: A Lost Opportunity or a New Beginning?, *The Cartographic Journal*, 50 (2), 166-170.

Mao, B., Wu, Z. ve Cao, J., 2012, A framework for online spatio-temporal data visualization based on HTML5, *Proceedings of the International Conference on Photogrammetry, Remote Sensing and Spatial Information Sciences,(XXII ISPRS Congress, Melbourne, Australia, 25 August 01 September, 2012)*, 123-127.

MapTiler, 2016, Tiles à la Google Maps: Coordinates, Tile Bounds and Projection, http://www.maptiler.org/google-maps-coordinates-tile-bounds-projection/: [20.02.2016].

Marbouti, M., Bhaskar, R., Shakeri, Z., Anslow, C., Jackson, L. ve Maurer, F., 2015, Designing Geovisual Analytics Application for Exploring Hydrological Data.

Marek, L., Tuček, P. ve Pászto, V., 2015, Using geovisual analytics in Google Earth to understand disease distribution: a case study of campylobacteriosis in the Czech Republic (2008–2012), *International Journal of Health Geographics*, 14 (1), 1.

Maso, J., Pons, X. ve Zabala, A., 2012, Tuning the second-generation SDI: theoretical aspects and real use cases, *International Journal of Geographical Information Science*, 26 (6), 983-1014.

May, A., Parker, C. J., Taylor, N. ve Ross, T., 2014, Evaluating a concept design of a crowd-sourced 'mashup' providing ease-of-access information for people with limited mobility, *Transportation Research Part C-Emerging Technologies*, 49, 103-113.

Michael, K., Bunch, M. ve Varma, S., 2013, Simulation of the cumulative impacts of CO2 geological storage and petroleum production on aquifer pressures in the offshore Gippsland Basin, *International Journal of Greenhouse Gas Control*, 19, 310-321.

Mielke, L. ve Burger, H., 2007, Thematic web-mapping potential and limitations of Google Maps API., *12th Conference of International Association for Mathematical Geology*, Beijing, China, 715-718.

Misra, A., Gooze, A., Watkins, K., Asad, M. ve Le Dantec, C., 2014, Crowdsourcing and its application to transportation data collection and management, *Transportation Research Record: Journal of the Transportation Research Board* (2414), 1-8.

Nelson, J. K., Quinn, S., Swedberg, B., Chu, W. H. ve MacEachren, A. M., 2015, Geovisual Analytics Approach to Exploring Public Political Discourse on Twitter, *Isprs International Journal of Geo-Information*, 4 (1), 337-366.

Nelson, S. M., Fielding, E. J., Zamora-Arroyo, F. ve Flessa, K., 2013, Delta dynamics: Effects of a major earthquake, tides, and river flows on Cienega de Santa Clara and the Colorado River Delta, Mexico, *Ecological Engineering*, 59, 144-156.

Newman, G., Zimmerman, D., Crall, A., Laituri, M., Graham, J. ve Stapel, L., 2010, User-friendly web mapping: lessons from a citizen science website, *International Journal of Geographical Information Science*, 24 (12), 1851-1869.

Ng, A. H. M., Ge, L. L. ve Li, X. J., 2015, Assessments of land subsidence in the Gippsland Basin of Australia using ALOS PALSAR data, *Remote Sensing of Environment*, 159, 86-101.

Nguyen, H. T., 2015, The Approach of Space-Time Cube to Visualizing Movement Data, *International Journal of Modeling and Optimization*, 5 (3), 207.

Nielsen, 2014, Tops of 2014: Digital, http://www.nielsen.com/us/en/insights/news/2014/tops-of-2014-digital.html: [19.01.2016].

Nittel, S., Bodum, L., C. Clarke, K., Gould, M., Raposo, P., Sharma, J. ve Vasardani, M., 2015, Emerging Technological Trends likely to Affect GIScience in the Next 20 Years.

Nuojua, J., 2010, WebMapMedia: a map-based Web application for facilitating participation in spatial planning, *Multimedia Systems*, 16 (1), 3-21.

O'Reilly, T., 2005, Web 2.0: compact definition, *Message posted to http://radar. oreilly. com/archives/2005/10/web_20_compact_definition. html*.

OSM_wiki, 2016, Slippy Map Tilenames http://wiki.openstreetmap.org/wiki/Slippy_map_tilenames: [20.02.2016].

Parker, C. J., May, A. ve Mitchell, V., 2014, User-centred design of neogeography: the impact of volunteered geographic information on users' perceptions of online map 'mashups', *Ergonomics*, 57 (7), 987-997.

Patel, A., Seyfi, A., Taghavi, M., Wills, C., Na, L., Latih, R. ve Misra, S., 2012, A Comparative Study of Agile, Component-Based, Aspect-Oriented and Mashup Software Development Methods, *Tehnicki Vjesnik-Technical Gazette*, 19 (1), 175-189.

Peck, R., Olsen, C. ve Devore, J., 2015, Introduction to statistics and data analysis, Cengage Learning, p.

Peterson, M. P., 2012a, Online mapping with APIs, In: Online Maps with APIs and WebServices, Eds: Springer, p. 3-12.

Peterson, M. P., 2012b, Online maps with APIs and WebServices, Springer Science & Business Media, p.

Peterson, M. P., 2015, Evaluating mapping APIs, In: Modern Trends in Cartography, Eds: Springer, p. 183-197.

Peuquet, D. J., Robinson, A. C., Stehle, S., Hardisty, F. A. ve Luo, W., 2015, A method for discovery and analysis of temporal patterns in complex event data, *International Journal of Geographical Information Science*, 29 (9), 1588-1611.

Pietroniro, E. ve Fichter, D., 2006, Map mashups and the rise of amateur cartographers and mapmakers, *BULLETIN-ASSOCIATION OF CANADIAN MAP LIBRARIES AND ARCHIVES*, 127, 26.

Plewe, B., 2007, Web cartography in the United States, *Cartography and Geographic Information Science*, 34 (2), 133-136.

Poorazizi, M. E. ve Alesheikh, A. A., 2011, Monitoring real-time environmental information using Web 2.0 and GIServices technology, *International Journal of Civil Engineering*, 9 (1), 63-70.

Programmableweb, 2016, Mashups by Category, http://www.programmableweb.com/category-mashups: [02.02.2016].

Proulx, P., Tandon, S., Bodnar, A., Schroh, D., Harper, R. ve Wright, W., 2006, Avian flu case study with nSpace and GeoTime, *Visual Analytics Science And Technology, 2006 IEEE Symposium On*, 27-34.

Putz, S., 1994, Interactive information services using World-Wide Web hypertext, *Computer Networks and ISDN Systems*, 27 (2), 273-280.

Raspini, F., Cigna, F. ve Moretti, S., 2012, Multi-temporal mapping of land subsidence at basin scale exploiting Persistent Scatterer Interferometry: case study of Gioia Tauro plain (Italy), *Journal of Maps*, 8 (4), 514-524.

Richter, F., 2013, Goggle Maps is the Most-Used Smartphone App in the World, https://www.statista.com/chart/1345/top-10-smartphone-apps-in-q2-2013/: [20.03.2016].

Roth, R. E., Robinson, A., Stryker, M., MacEachren, A. M., Lengerich, E. J. ve Koua, E., 2008, Web-based geovisualization and geocollaboration: Applications to public health, *Proceedings of the Joint Statistical Meeting, Invited Session on Web Mapping*.

Roth, R. E., Donohue, R. G., Sack, C. M., Wallace, T. R. ve Buckingham, T. M. A., 2015, A Process for Keeping Pace with Evolving Web Mapping Technologies, *Cartographic Perspectives*, 0 (78), 25-52.

Schmidt, M. ve Weiser, P., 2012, Web mapping services: Development and trends, In: Online Maps with APIs and WebServices, Eds: Springer, p. 13-21.

Singh, P. S., Chutia, D. ve Sudhakar, S., 2012, Development of a Web Based GIS Application for Spatial Natural Resources Information System Using Effective Open Source Software and Standards, *Journal of Geographic Information System*, 04 (03), 261-266.

Steed, C. A., Ricciuto, D. M., Shipman, G., Smith, B., Thornton, P. E., Wang, D. L., Shi, X. Y. ve Williams, D. N., 2013, Big data visual analytics for exploratory earth system simulation analysis, *Computers & Geosciences*, 61, 71-82.

Sukic, E. ve Rancic, D., 2011, FOS GIS solutions in Web Mapping, *Technics Technologies Education Management-Ttem*, 6 (4), 1083-1088.

Sukic, E. ve Stoimenov, L., 2012, The Use of Web Mapping Services for Designing e-City Government Web Portals, *Technics Technologies Education Management-Ttem*, 7 (2), 875-881.

Teatini, P., Castelletto, N., Ferronato, M., Gambolati, G., Janna, C., Cairo, E., Marzorati, D., Colombo, D., Ferretti, A., Bagliani, A. ve Bottazzi, F., 2011, Geomechanical response to seasonal gas storage in depleted reservoirs: A case study in the Po River basin, Italy, *Journal of Geophysical Research-Earth Surface*, 116.

Teatini, P., Castelletto, N. ve Gambolati, G., 2014, 3D geomechanical modeling for CO2 geological storage in faulted formations. A case study in an offshore northern Adriatic reservoir, Italy, *International Journal of Greenhouse Gas Control*, 22, 63-76.

Ternay, G. d., 2015, Business Ecosystem: Creating an Economic Moat, https://boostcompanies.com/business-ecosystem/: [17.02.2016].

Thom, D., Bosch, H., Koch, S., Worner, M. ve Ertl, T., 2012, Spatiotemporal Anomaly Detection through Visual Analysis of Geolocated Twitter Messages, *Ieee Pacific Visualization Symposium 2012*, 41-48.

Turdukulov, U. ve Moncrieff, S., 2015, Enabling geovisual analytics of health data using a server-side approach, *Cartography and Geographic Information Science*, 43 (1), 16-29.

Turnbull, B. ve Randhawa, S., 2015, Automated event and social network extraction from digital evidence sources with ontological mapping, *Digital Investigation*, 13, 94-106.

Ustun, A., Tusat, E. ve Yalvac, S., 2010, Preliminary results of land subsidence monitoring project in Konya Closed Basin between 2006–2009 by means of GNSS observations, *Natural Hazards and Earth System Sciences*, 10 (6), 1151-1157.

Van Harmelen, F., Lifschitz, V. ve Porter, B., 2008, Handbook of knowledge representation, Elsevier, p.

Wagner, J., 2015, Top 10 Mapping APIs: Google Maps, Microsoft Bing Maps and MapQuest, http://www.programmableweb.com/news/top-10-mapping-apis-google-maps-microsoft-bing-maps-and-mapquest/analysis/2015/02/23: [17.02.2016].

Weiss, M. ve Gangadharan, G., 2010, Modeling the mashup ecosystem: structure and growth, *R&d Management*, 40 (1), 40-49.

Werts, J. D., Mikhailova, E. A., Post, C. J. ve Sharp, J. L., 2012a, An integrated WebGIS framework for volunteered geographic information and social media in soil and water conservation, *Environmental Management*, 49 (4), 816-832.

Werts, J. D., Mikhailova, E. A., Post, C. J. ve Sharp, J. L., 2012b, An Integrated WebGIS Framework for Volunteered Geographic Information and Social Media in Soil and Water Conservation, *Environmental Management*, 49 (4), 816-832.

Wikipedia, 2016a, Web Mapping in Wikipedia, https://en.wikipedia.org/wiki/Web_mapping: [12.01.2016].

Wikipedia, 2016b, Web 2.0 In Wikipedia https://en.wikipedia.org/wiki/Web_2.0#.22Web_1.0.22: [29.01.2016].

Wikipedia, 2016c, Mashup (web application hybrid) in Wikipedia, https://en.wikipedia.org/wiki/Mashup_(web_application_hybrid): [22.01.2016].

Wiringa, A. E., Shutt, K. A., Marsh, J. W., Cohn, A. C., Messonnier, N. E., Zansky, S. M., Petit, S., Farley, M. M., Gershman, K., Lynfield, R., Reingold, A., Schaffner, W., Thompson, J., Brown, S. T., Lee, B. Y. ve Harrison, L. H., 2013,

Geotemporal analysis of Neisseria meningitidis clones in the United States: 2000-2005, *Plos One*, 8 (12), e82048.

Witten, I. H. ve Frank, E., 2005, Data Mining: Practical machine learning tools and techniques, Morgan Kaufmann, p.

Yvonne, R., Sharp, H. ve Preece, J., 2011, Interaction Design: Beyond Human-Computer Interaction, Wiley Publishing.

Zhang, D., Chow, C.-Y., Li, Q., Zhang, X. ve Xu, Y., 2012, SMashQ: spatial mashup framework for k-NN queries in time-dependent road networks, *Distributed and Parallel Databases*, 31 (2), 259-287.

Zhang, T., Hua, G. F. ve Ligmann-Zielinska, A., 2015, Visually-driven parallel solving of multi-objective land-use allocation problems: a case study in Chelan, Washington, *Earth Science Informatics*, 8 (4), 809-825.

Zhu, W., Zhang, Q., Ding, X. L., Zhao, C. Y., Yang, C. S. ve Qu, W., 2013, Recent ground deformation of Taiyuan basin (China) investigated with C-, L-, and X-bands SAR images, *Journal of Geodynamics*, 70, 28-35.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı	**:**	Osman Sami KIRTILOĞLU
Uyruğu	**:**	T.C.
Doğum Yeri ve Tarihi	**:**	Erzincan / 1983
Telefon	**:**	+90 530 433 38 30
Faks	**:**	+90 332 241 06 35
e-mail	**:**	okirtiloglu@selcuk.edu.tr – osmankirtiloglu@gmail.com

EĞİTİM

Derece		**Adı, İlçe, İl**	**Bitirme Yılı**
Lise	:	Erzincan Anadolu Lisesi, Merkez, Erzincan	2001
Üniversite	:	Selçuk Üniversitesi	2005
Yüksek Lisans	:	Selçuk Üniversitesi	2009
Doktora	:	Selçuk Üniversitesi	2016

İŞ DENEYİMLERİ

Yıl	**Kurum**	**Görevi**
2001	Selçuk Üniversitesi Fen Bilimleri Enstitüsü	Araştırma Görevlisi

UZMANLIK ALANI

Kartografya, Harita Projeksiyonları, Web Kartografya, Coğrafi Bilgi Sistemleri, Tematik Kartografya, Web Harita Servisleri, HTML, CSS, JavaScript

YABANCI DİLLER

İngilizce

YAYINLAR

Üstün A., Tuşat E., Yalvaç S., Özkan İ., Eren Y., Özdemir A., Bildirici İ. Ö., Üstüntaş T., Kırtıloğlu O. S., Mesutoğlu M., Doğanalp S., Fatma C., Abbak R. A., Avşar N. B., Şimşek F. F., 2015, Land subsidence in Konya Closed Basin and its spatio-temporal detection by GPS and DInSAR. Environmental Earth Sciences, Doi: 10.1007/s12665-014-3890-5

Kırtıloğlu O. S., Bildirici İ. Ö., Uluğtekin N. N., Üstün A., 2015, Basin Monitoring System With Map Mash-Ups: Case Study of Konya Closed Basin. ICA European Symposium on Cartography.

Bildirici İ. Ö., Kırtıloğlu O. S., Uluğtekin N. N., 2014, Google Maps API and Google Earth API Integration: Development of an Educational Tool For Cartography Courses. 5th International Conference on Cartography and GIS, 1, 433-443.

Kırtıloğlu O. S., Bildirici İ. Ö., Üstün A., 2014, Developing Geovisual Analytics Environment With Map Mashups: A Case Study of Konya Closed Basin. 5th International Conference on Cartography and GIS, 1, 498-507.

Bildirici İ. Ö., Kırtıloğlu O. S., Uluğtekin N. N., 2013, Using Map Mashups in Map Projection Education. 26 International Crtographic Conference, 1, 72