

# Örün Tabanlı Coğrafi Bilgi Sistemi Uygulama Programlama Arayüzü Geliştirilmesi

Bülent ÖZHORASAN<sup>1</sup>, Hüseyin ATEŞ<sup>2</sup>

<sup>1</sup> Havelsan A.Ş. , Ankara, bozhorasan@havelsan.com.tr

<sup>2</sup> Havelsan A.Ş. , Ankara, hates@havelsan.com.tr

**Özet** Örün tarayıcılarında herhangi bir kurulum gerektirmeden çalışacak Coğrafi Bilgi Sistemi uygulamalarına olan ihtiyaç artmaktadır. Bu kapsamda WebGL ve JavaScript kullanılarak geliştirilen TMAP-W (Tactical Map-Web) ürünü örün tabanlı CBS uygulamalarında CBS altyapısı olarak kullanılabilir. Bu bildiriye uygulama programlama arayüzünün tasarımı, gerçekleştirimi ve inşa süreci hakkında bilgi verilecektir.

**Anahtar Kelimeler:** Örün Tabanlı Coğrafi Bilgi Sistemi, JavaScript, Uygulama Programlama Arayüzü

## 1 Giriş

TMAP-W (Tactical Map – Web), örün tabanlı coğrafi bilgi sistemi uygulamalarında altyapı olarak kullanmak için Havelsan tarafından geliştirilen bir UPA (Uygulama Programlama Arayüzü)'dür. Java programlama dilinde geliştirilmiş TMAP-J in Uygulama Programlama Arayüzü ile aynı arayüze sahiptir. TMAP-W, kullanılan CBS aracından bağımsız olarak tasarlanmıştır. Bir başka deyişle, CBS aracına özgü gerçekleştirme kısımları yeniden yazılarak TMAP-W ürünü kullanan projelerin kodlarında değişiklik yapmadan CBS aracı değişikliğine gidilebilir.

Ayrıca CBS aracının sürüm güncellemeleri ve her türlü değişiklikleri de TMAP-W kullanan projelere yansıtılmadan TMAP-W kodunda ele alınır.

TMAP-W kendisini CBS altyapısı olarak kullanacak projenin teknoloji seçiminden de bağımsızdır, tek gerek ve yeter koşul, seçilen teknolojinin örün tarayıcısı üzerinde çalışıyor olmasıdır. Örneğin Apache Wicket veya Asp.Net gibi çerçevelerle kullanılabilir.

Bu makalede TMAP-W ürününün temel tasarım unsurları, ehlileştirilmiş JavaScript yaklaşımı ve inşa yaklaşımı hakkında deneyimler aktarılacaktır.

## 2 Temel Tasarım Unsurları

### 2.1 JavaScript Dilinin Seçimi

Örün tabanlı bir uygulamada Java kodları applet kullanılarak örün tarayıcısı içinden kullanılabilir. Ancak bunun için istemciye JRE (Java Runtime Environment) kurulması ve tarayıcıda Java'nın etkinleştirilmesi gerekmektedir. Bu durumda tarayıcıda güvenlik açıkları oluşmaktadır<sup>[1]</sup>. Ayrıca Appletlerin sertifikalarının zamanını doldurması gibi sorunlar da oluşabilmekte sertifikaların yenilenmesi gibi işlemler de kullanıcının yazılımın kullanılabilirliğine olan algısını olumsuz yönde

değiştirebilmektedir. Bu olumsuz yönlerinden dolayı gerçekleştirilmede TMAP-J ve Java programlama dili kullanılmamıştır.

JavaScript, dinamik ve prototip tabanlı bir betik dilidir. JavaScript fonksiyonları bir değişken olarak kullanabilmeye olanak sağlamaktadır. JavaScript EcmaScript ile standartlaştırılmıştır<sup>[2]</sup> ve ürün tarayıcılarının tümü tarafından desteklenmektedir. Bu özellik yüzünden JavaScript ürün tabanlı bir CBS çözümünün tüm tarayıcılarla uyumlu olması için tek çözüm olarak ortaya çıkmaktadır. Yukarıda açıklanan nedenlerden dolayı JavaScript geliştirme dili olarak seçilmiştir.

## 2.2 WebGL ve Cesium

WebGL platformlar arası, lisans gerektirmeyen alt seviyede 3B (üç boyutlu) ve 2B (iki boyutlu) grafik UPA dır. OpenGL, ES 2.0 standardı üzerine tanımlanmıştır. WebGL, ürün uygulamalarına tarayıcıya eklenti kurmadan 3B özelliği kazandırılmasına olanak sağlar<sup>[3]</sup>. WebGL' i ürün tarayıcıların en yeni sürümleri desteklemektedir ve kullanımı gittikçe yaygınlaşmaktadır.

WebGL 'in grafik işlemciyi yoğun olarak kullanmasından dolayı, WebGL uygulamalarının yüksek başarılı çalışması için donanım hızlandırması gerekmektedir. Bu nedenle paylaşımli değil ayrı grafik kartı kullanımı daha uygun olmaktadır.

Yukarıda bahsedilen avantajlarına rağmen, WebGL ile geliştirme maliyetli ve karmaşıktır. Assembly dili ile masaüstü yazılım geliştirme ile karşılaştırılabilir. Bu nedenle WebGL üzerine CBS ile ilgili özellikleri gerçekleyen bir kütüphane arayışı oluşmuş ve bu arayışın sonunda Cesium<sup>[4]</sup> ortaya çıkmıştır.

Cesium, 2B ve 3B haritalar yaratmak için JavaScript ve WebGL tabanlı, ürün tarayıcısında bir eklenti gerektirmeyen bir kütüphanedir. Açık kaynaklı ve Apache 2.0 lisansına sahip olduğu için kullanılması ücretsizdir.

Cesium, Patrick Cozzi liderliğinde bir ekip tarafından geliştirilmektedir. Patrick Cozzi'nin bilgisayar grafik ve Grafik İşleme Birimi (GPU) programlanması ve 3B Küresel Görselleştirme hakkında çalışmaları vardır<sup>[5]</sup>.

## 2.3 TMAP-J UPA Benzetimi

TMAP-J, Java dilinde OpenGL kullanılarak geliştirilmiş bir CBS altyapısıdır. Havelsan 'da çeşitli projeler tarafından kullanılmaktadır. TMAP-W 'in tasarım aşamasında TMAP-J 'in UPA seçilerek JavaScript'e taşınmasına karar verilmiştir. Bu sayede TMAP-J ve Java kullanarak masaüstü uygulama geliştiren geliştiricilerin ürün uygulaması geliştirirken tanıdık oldukları UPA kullanmaları sağlanmış ve TMAP-J ve TMAP-W 'in aynı arayüzü değişik dillerde gerçekleyen ürünler olmaları sağlanmıştır. Sonuç olarak geliştiriciler yeni bir UPA öğrenmek zorunda kalmadan Java ve JavaScript 'de CBS uygulamaları geliştirebileceklerdir.

### 3 Ehlileştirilmiş JavaScript Yaklaşımı

JavaScript 'in tip denetlemesi yapmaması ve nesneye yönelik programlama ile ilgili sınıf, miras alma gibi özelliklerin henüz bulunmaması JavaScript geliştirmesi için bazı araçlar ve yaklaşımlar kullanılarak nesneye yönelik programlamayı destekleyen dillere (Örneğin Java, C#) yakınsatılması ihtiyacını doğurmaktadır. Bu yaklaşıma kısaca ehlileştirilmiş JavaScript denmektedir <sup>[6]</sup>.

#### 3.1 JavaScript ile Nesneye Yönelik Programlama

TMAP-W, EcmaScript standardının Haziran 2011'de yayınlanan 5.1 sürümü ile uyumlu geliştirilmiştir. 2015'in ortasında yayınlanması planlanan sürüm 6, sınıfları ve modülleri destekleyecektir. JavaScript dilinin desteklediği EcmaScript standardının güncel sürümü olan 5.1, nesneye yönelik programlama özelliklerini barındırmamaktadır.

Nesneye yönelik programlama özelliklerinin JavaScript 'de kullanılması için açık kaynak Google Closure kütüphanesi kullanılmıştır.

**Error! Reference source not found.**'de örnek sınıf olarak çember nesnesi IShape arayüzünden goog.inherits komutu ile türemektedir. Closure ile çoklu miras alma da goog.mixin ile mümkün olmaktadır<sup>[7]</sup>.

```
1  /*global IShape,goog,console*/
2  /* @constructor
3   * @Implements IShape
4   */
5  function Circle() {
6     /*@type number*/
7     var radius = 0;
8
9     goog.base(this);
10
11    goog.inherits(Circle, IShape);
12
13    Circle.prototype.draw = function () {
14        console.log('Circle draw implementation..');
15    };
16
17    /**
18     * Calculates area of circle
19     * @return {number} Calculated area
20     */
21    Circle.prototype.calculateArea = function () {
22        return Math.PI * Math.pow(this.radius, 2);
23    };
24
25    /**
26     * Enlarges the circle radius by a factor
27     * @param factor number Enlargement Factor
28     */
29    Circle.prototype.enlarge = function (factor) {
30        this.radius = this.radius * factor;
31    };

```

Şekil 1 Çember Sınıfının Tanımlanması

### 3.2 Gömülü Örnekler Yaklaşımı

TMAP-W içinde her özelliğin nasıl kullanılacağına dair çalışan bir kod örneğinin bulunduğu Gömülü Örnekler uygulaması bulunmaktadır. Bu sayfaya TMAP-W geliştiricileri tarafından ekleme ve düzeltmeler yapılmakta, aynı sayfa test grubu tarafından da geçirme ve doğrulama faaliyetlerinde araç olarak kullanılmaktadır. Aynı zamanda ürünü kullanan geliştiriciler tarafından da referans olarak kullanılmaktadır. Alınan her sürümde gömülü örnekler sayfasına yeni örnekler eklenebilmekte, var olan örnekler test grubu ve müşterilerden gelen geri beslemelere göre genişletilmektedir. **Error! Reference source not found.**'de bir TMAP-W sürümünün gömülü örnekler sayfası görülmektedir.

#### TMAP Web 150301-01 Examples

[Click Here for Unit Tests](#)

NO	NAME	EXPLANATION	CODE	REQUIREMENT ID	TEST READY
001	Set Center	Setting the center point of the map	Code	1000	Yes
002	Zoom In	Zoom In	Code	1001	Yes
003	Zoom Out	Zoom Out	Code	1002	Yes
004	Mouse Location	Mouse Location	Code	1003	No
005	Key Location	Key Location	Code	1004	Yes
006	Great Circle and Rhumbline Distance Measurement	Great Circle and Rhumbline Distance Measurement	Code	1005	Yes
007	Graphic Label	Graphic Label	Code	1006, 1007, 1008	Yes
008	Graphic Item	Graphic Item	Code	1009	Yes
009	Layer Location	Layer Location	Code	1010	Yes
010	Applet Item	Applet Item	Code		Yes
011	Graphic Line String	Graphic Line String	Code		Yes
012	Polygon	Polygon	Code		Yes
013	More Countries	More Countries	Code	1011	Yes
014	Elevation Service	Elevation Service	Code	1012, 1013	Yes
015	Performance	Performance	Code		Yes
016	Shapefile Decoder	Shapefile Decoder Polygon	Code, 1014	1014	Yes
017	APPN File	APPN File	Code	1015, 1016	Yes
018	Virtual File	Virtual File To Server	Code		Yes
019	Shapefile Decoder Item	Shapefile Decoder Item	Code	1017	Yes
020	Shapefile Decoder Line	Shapefile Decoder Line	Code	1018	Yes
021	Shapefile Decoder Line	Shapefile Decoder Line	Code	1019	Yes
022	Circle Class	Creating circle...	Code		No

Şekil 2 TMAP-W Gömülü Örnekler Sayfası

### 3.3 JavaScript Birim Testleri

TMAP-W içerisinde hesaplamaya dayalı jeodezik metotlar bulunmaktadır. Bu metotların birim testi için bir çerçeveye ihtiyaç duyulmuştur. Bunun için JUnit benzeri bir yapıya sahip QUnit<sup>[8]</sup> seçilmiştir. TMAP-W projesi içinde 300den fazla birim testi her sürüm alımı öncesi çalıştırılmakta ve başarıyla geçtikleri teyit edildikten sonra sürüm çıkarılmaktadır. **Error! Reference source not found.**'de TMAP-W birim testleri sayfası bulunmaktadır. Sayfada her test grubundaki parantez içindeki değerler sırasıyla hata alan test sayısını, geçen test sayısını ve toplam test sayısını belirtmektedir.

TMAP Web QUnit Unit Tests		
<input type="checkbox"/> Hide passed tests	<input type="checkbox"/> Check for Globals	<input type="checkbox"/> No try-catch
Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/33.0.1750.154 Safari/537.36		
Tests completed in 106 milliseconds. 329 assertions of 329 passed, 0 failed.		
1. Sample Test	(0, 1, 1)	Rerun
2. Great Circle Distance Test	(0, 46, 46)	Rerun
3. Rhumb Line Distance Test	(0, 46, 46)	Rerun
4. Great Circle Bearing Test	(0, 23, 23)	Rerun
5. Rhumb Line Bearing Test	(0, 45, 45)	Rerun
6. Great Circle Reckon Test	(0, 46, 46)	Rerun
7. Rhumb Line Reckon Test	(0, 92, 92)	Rerun
8. Great Circle Path Test	(0, 3, 3)	Rerun
9. Rhumb Line Path Test	(0, 3, 3)	Rerun
10. WGS84 Parameters Test	(0, 2, 2)	Rerun
11. Intersection Points	(0, 4, 4)	Rerun
12. Reckon	(0, 6, 6)	Rerun
13. Coordinate Transformation	(0, 12, 12)	Rerun

Şekil 3 TMAP-W QUnit Birim Testleri

## 4 İnşa Yaklaşımı

TMAP-W geliştirimi 2 haftalık yinelemelerde geliştirilecek ihtiyaçların planlanması ve hataların çözülmesi şeklinde olmaktadır. Her 2 haftalık yineleme sonunda bir sürüm alınmaktadır. Geliştirme Planı'nda belirlenen sıklıklarda, resmi sürüm alınan ara sürümler arasından seçilip test grubu tarafından geçерleme ve doğrulaması yapılmaktadır.

✓ TMAPW-JS\_20150318.1 - Derleme başanlı

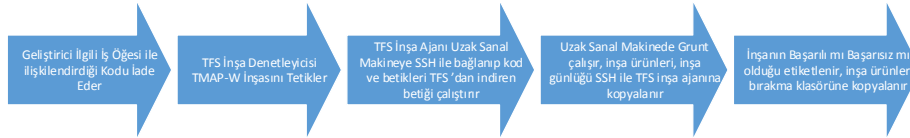
Özet **Günlük** Tanılama

Bırakma klasörünü aç | Sonsuza kadar koru | ✗ | <Hiçbir kalite atanmadı>

Serpil DİKMEN TMAPW-JS (KKSS) ögesini 22129 değişiklik kümesi için tetikledi  
21.2 saat önce 3.4 dakika (KKSS) için tamamlanan çalışma,

```
Running "clean:0" (clean) task
Cleaning dist...OK
Running "clean:1" (clean) task
Cleaning doc...OK
Running "jslintclient" (jslint) task
FAIL:source5/tr/com/havelsan/tmap/api/datasource/gisdecoder.js (4)
source5/tr/com/havelsan/tmap/api/datasource/gisdecoder.js:8:30:Empty block.
source5/tr/com/havelsan/tmap/api/datasource/gisdecoder.js:8:22:Unused 'params'.
source5/tr/com/havelsan/tmap/api/datasource/gisdecoder.js:26:47:Unused 'success'.
source5/tr/com/havelsan/tmap/api/datasource/gisdecoder.js:26:56:Unused 'fail'.
FAIL:source5/tr/com/havelsan/tmap/api/datasource/gisdecoderfactory.js (6)
source5/tr/com/havelsan/tmap/api/datasource/gisdecoderfactory.js:46:66:Unused 'provider'.
source5/tr/com/havelsan/tmap/api/datasource/gisdecoderfactory.js:46:66:Unused 'params'.
source5/tr/com/havelsan/tmap/api/datasource/gisdecoderfactory.js:46:74:Unused 'format'.
source5/tr/com/havelsan/tmap/api/datasource/gisdecoderfactory.js:60:64:Unused 'data'.
source5/tr/com/havelsan/tmap/api/datasource/gisdecoderfactory.js:60:70:Unused 'params'.
source5/tr/com/havelsan/tmap/api/datasource/gisdecoderfactory.js:60:78:Unused 'format'.
FAIL:source5/tr/com/havelsan/tmap/api/datasource/gisvectordecoder.js (1)
source5/tr/com/havelsan/tmap/api/datasource/gisvectordecoder.js:8:30:Empty block.
FAIL:source5/tr/com/havelsan/tmap/api/datasource/giswebwfsccontroller.js (1)
source5/tr/com/havelsan/tmap/api/datasource/giswebwfsccontroller.js:8:33:Empty block.
FAIL:source5/tr/com/havelsan/tmap/api/geometry/defaultgisgeodesy.js (6)
source5/tr/com/havelsan/tmap/api/geometry/defaultgisgeodesy.js:20:10:29:Unexpected dangling '_' in '_checkLineIntersection_'.
source5/tr/com/havelsan/tmap/api/geometry/defaultgisgeodesy.js:20:85:13:Unexpected 'continue'.
source5/tr/com/havelsan/tmap/api/geometry/defaultgisgeodesy.js:20:94:17:Unexpected 'continue'.
source5/tr/com/havelsan/tmap/api/geometry/defaultgisgeodesy.js:20:98:17:Don't declare variables in a loop.
source5/tr/com/havelsan/tmap/api/geometry/defaultgisgeodesy.js:20:98:31:Unexpected dangling '_' in '_checkLineIntersection_'.
source5/tr/com/havelsan/tmap/api/geometry/defaultgisgeodesy.js:21:64:17:Don't declare variables in a loop.
FAIL:source5/tr/com/havelsan/tmap/api/geometry/gisposition.js (1)
source5/tr/com/havelsan/tmap/api/geometry/gisposition.js:146:13:Expected exactly one space between 'return' and 'points'.
FAIL:source5/tr/com/havelsan/tmap/api/geometry/graphicoperations.js (1)
source5/tr/com/havelsan/tmap/api/geometry/graphicoperations.js:3:31:Empty block.
```

Şekil 4 İnşa Günlüğü Detayı – JSLint Uyarıları



Şekil 5 TMAP-W İnşa Süreci

#### 4.1 JavaScript Projelerinde İnşa

JavaScript projelerinde inşa aracı olarak açık kaynak Grunt<sup>[9]</sup> aracının kullanımı yaygındır. TMAP-W projesinde de Grunt aracı kullanılarak kodların birleştirilmesi, hata kontrolü ve dokümantasyon üretimi yapılmaktadır. **Error! Reference source not found.**'de TFS inşa günlüğüne JsLint<sup>[10]</sup> uyarıları tümleştirilmiş olarak gösterilmiştir. JsLint aracı hem TGO (Tümleşik Geliştirme Ortamı) içinde hem de inşa sürecinde kod denetleyici olarak kullanılmaktadır.

## 4.2 JavaScript İnşalarının Team Foundation Server (TFS) İnşa Birimi ile Tümüleştirilmesi

Havelsan 'da Uygulama Yaşam Döngüsü Yönetimi aracı olarak Microsoft Team Foundation Server sürüm 12 (TFS)<sup>[11]</sup> kullanılmaktadır. Bu aracın sürekli tümleştirme ve inşa özellikleri de mevcuttur.

TFS 'in JavaScript projelerinin inşası ile ilgili hazır bir çözümü bulunmamaktadır. Ancak esnek bir özelleştirme altyapısı mevcuttur.

Grunt inşa aracı, SSH hizmeti etkinleştirilmiş bir sanal makinede çalışmaktadır. Sanal makinede JavaScript inşasını başlatacak betik dosyaları bulunmaktadır. TFS İnşa Ajansı SSH vasıtasıyla uzaktan bu betik dosyalarını çalıştırmakta, inşa kayıt dosyası ve inşa ürünleri de TFS inşa rehberlerine kopyalanmaktadır. Ayrıca betik dosyasının hata döndürüp döndürmediğine göre inşanın başarılı olduğu veya hatalı olduğu TFS tarafında kurulmaktadır. Şekil 5'de inşa süreci görülebilir.

## 4.3 JsDoc ile Projenin Dokümantasyonunun Üretilmesi

JsDoc<sup>[12]</sup>, JavaScript projelerinin dokümantasyonunun otomatik olarak üretilmesi için yaygın olarak kullanılan bir araçtır. Hazırlanan UPA 'nın dokümantasyonunun üretilmesi için JsDoc kullanılmıştır. İnşa sürecine tümleştirilen JsDoc üretimi sayesinde, her inşa sonrası dokümantasyon yeniden üretilmekte ve böylece kod ve dokümantasyonun tutarlı kalması sağlanmaktadır.

## 5 Sonuç

Örün Tabanlı Coğrafi Bilgi Sistemi Uygulama Programlama Arayüzü Geliştirilmesi kapsamında internet tarayıcılarında bir eklenti kurulumu gerektirmeden çalışan TMAP-W ürününün geliştirilmesi hakkında tasarım kararları, ehliştirilmiş JavaScript yaklaşımı ve inşa yaklaşımı ile ilgili deneyimler aktarılmıştır.

TMAP-W ürününe yeni özellikler eklenmeye devam edilmekte ve Havelsan içinde projelerin altyapı olarak kullanması da yönetim tarafından teşvik edilmektedir.

## Kaynaklar

1. Java Runtime Environment Perpetual Vulnerability Machine <http://www.f-secure.com/weblog/archives/00002285.html>. Görüntülenme Tarihi 17 Nisan 2015.
2. ECMAScript Language Specification, Edition 5.1 Haziran 2011. <http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>. Görüntülenme Tarihi 1 Nisan 2015
3. WebGL, OpenGL ES 2.0 for the Web. <https://khronos.org/webgl>. Görüntülenme Tarihi 1 Nisan 2015
4. Cesium WebGL Virtual Globe and Map Engine. <http://cesiumjs.org>. Görüntülenme Tarihi 1 Nisan 2015
5. Cozzi P., Ring K. (2011), 3D Engine Design for Virtual Globes, 1<sup>st</sup> Edition. CRC Press, Boca Raton.

6. Wei S., Ryder G. B. (2014) Taming the Dynamic behaviour of JavaScript, SPLASH'14 Proceedings of the 2014 ACM SIGPLAN conference on Systems, Programming, and Applications: Software for Humanity.
7. Bolin M. (2010), Closure The Definitive Guide, 1<sup>st</sup> Edition. O'Reilly, Beijing.
8. QUnit <https://qunitjs.com>. Görüntülenme Tarihi 17 Nisan 2015.
9. Grunt <http://www.gruntjs.com>. Görüntülenme Tarihi 17 Nisan 2015
10. JSLint <http://www.jshint.com>. Görüntülenme Tarihi 17 Nisan 2015
11. Team Foundation Server <http://www.visualstudio.com/tr-tr/products/tfs-overview-vs.aspx>
12. JSDoc <https://github.com/jsdoc3/jsdoc>. Görüntülenme Tarihi 17 Nisan 2015.