

UKVA Kapsamında UVDF-GML Tasarımı

Ozan EMEM¹, Fatmagül BATUK², İsmail Ercüment AYAZLI³

Özet

Büyük Ölçekli Harita ve Harita Bilgileri Üretim Yönetmeliği (BÖHHBÜY), harita üretimi ve jeodezik altyapının yanı sıra Coğrafi Bilgi Sistemlerinde (CBS) kullanılan standartların en önemlilerinden biridir. Veri paylaşımında kullanılabilen Detay Öznitelik Kodlama Kataloğu (DÖKK) ve Ulusal Veri Değişimi Formatı (UVDF) bu yönetmelik ile geliştirilmiştir. Başta kamu projeleri olmak üzere günümüzde pek çok projede UVDF kullanılmaktadır.

Bir kurumun Coğrafi Bilgi Sisteminin veri yapısı ya da daha geniş olarak Türkiye Ulusal Coğrafi Bilgi Sistemi (TUCBS) düşünüldüğünde, mevcut UVDF'nin yeterli olup olamayacağı bu makaleye konu olan araştırma ile incelenmiş, UVDF'nin uluslararası standart olan GML temelli bir yaklaşım ile modernleştirilebileceği, makalede örnek bir çalışma ile açıklanmıştır.

Anahtar Sözcükler

Birlikte çalışabilirlik, UVDF, GML, Standart, Veri değişimi

Abstract

Designing a National Data Exchange Format based on GML within Scope of National Spatial Data Infrastructure

Regulations for Production of Large Scale Map and Map Attributes is one of the important standards used for Geographical Information Systems (GIS) as well as map productions and geodetic infrastructures. The regulation also contains Feature Attribute Coding Catalogues (FACC) and National Data Exchange Format (NDEF). Nowadays, the NDEF has been used for data exchange in several public projects.

This paper discusses whether the present NDEF is adequate or not when considered the GIS data structure of an agency or even in larger scale the Turkish National Geographical Information System. Also, a modernized NDEF-GML approach has been introduced which is constituted according to the ISO 19136 GML International Standard.

Key Words

Interoperability, NDEF, GML, Standard, Data Exchange

1. Giriş

Uluslararası Standartlar Teşkilatının (International Organization for Standardization, ISO) 211 numaralı Coğrafi Bilgi/Geomatik Teknik Komitesi ve Açık CBS Konsorsiyumu (Open Geospatial Consortium, OGC) tarafından geliştirilen coğrafi veriler ve

yazılımlara yönelik standartlar CBS ve verilere yeni anlamlar ve yönler katmıştır (Emem, 2007). Standartlar, teknolojik gelişmeler, bilgi değişimi ve paylaşımında internetin kullanımı, verilerin değişiminin kullanıcıya fark ettirilmeden gerçekleştirilebildiği uygulamaları da ortaya çıkarmıştır (Chang-Tien vd, 2007).

Bir yazılım formatında depolanmış grafik verilerin, bir başka yazılım tarafından okunabilmesi için veri değişim formatı (Data Exchange Format, DXF) vb formatlar uzun yıllar kullanılmıştır. Teknolojinin daha da gelişmesi ile genişletilebilir işaretleme dili (eXtensible Markup Language, XML) tabanlı formatlar bu işlemleri daha da kolaylaştırmış ve taşınan verilere de zenginlik katmıştır. İşaretler, kurallar ve gramerler kümesi olan XML, veri paylaşımı ve veri değişim formatı açısından güçlü ve esnek bir yapı göstermektedir (URL 1).

2005 yılında yayınlanan BÖHHBÜY ile birlikte UVDF formatı geliştirilmiştir. UVDF, DÖKK dikkate alınarak özgün bir şema ile XML tabanlı olarak hazırlanmış ve yönetmelik eki olarak yayımlanmıştır (Çelik, 2007). DÖKK, büyük ölçekli harita verileri için detaylar, öznitelikler ve detay-öznitelik eşleştirmelerini içeren yazılı bir metindir.

Coğrafi veri işaretleme dili (Geography Markup Language, GML) ve ölçeklenebilir vektör grafikleri (Scalable Vector Graphics, SVG) coğrafi veriler ile ilişkili XML temelli uygulamalardır. GML, OGC tarafından geliştirilmiş ve 2007 yılında da ISO tarafından 19136 uluslararası standardı olarak kabul edilmiştir. Güncel sürümü 3.2.1 olan GML, kavram olarak temel şema, uygulama şemaları ve veri dokümanlarını içerir.

Günümüzde bir çok ülkede GML tabanlı ulusal veri değişim formatı oluşturulması çalışmaları yürütülmektedir. Ülkemizde uzun süredir gündemde olan e-Devlet çalışmaları kapsamında, Kamu Yönetiminde Modernizasyon-75 numaralı eylem ile, TUCBS'nin, dolayısıyla UKVA'nın geliştirilmesi hedeflenmektedir. UVDF'nin modernleştirilmesi de bu eylemin kapsamı içindedir.

Bu makalede, BÖHHBÜY DÖKK ve UVDF ile uyumlu olan bir GML uygulama şemasının tasarımı ve geliştirilmesi açıklanmıştır.

2. Coğrafi Verilerin Paylaşımı-Birlikte İşlerlik ve GML

1980'li yıllardan günümüze dek kurulmuş ve içinde değişik dosya yapılarında, formatlarda milyonlarca veri barındıran

¹ Dr. Araştırma Görevlisi, YTÜ, Jeodezi ve Fotogrametri Müh. Bölümü, İstanbul

² Doç. Dr., YTÜ, Jeodezi ve Fotogrametri Müh. Bölümü, İstanbul.

³ Araştırma Görevlisi, YTÜ, Jeodezi ve Fotogrametri Müh. Bölümü, İstanbul.

bir çok CBS vardır. Benzer şekilde ücretsiz yazılımların yanı sıra CBS'nin yazılım bileşenlerini ve dolayısıyla veri depolama formatlarını da geliştiren bir çok ticari firma bulunmaktadır. Bir çok firma kendine ait format geliştirmektedir; teknik açıdan bakıldığında bunun en büyük nedeni yazılımların, sistemlerde yapılacak sorgulamalar, analizler vb için sürekli veriler ile diyalogda olacak olmasıdır. Günümüzde CBS yazılımları bir çok formatı okuyabilmekte ve yazabilmektedir. Ancak; bir CBS'de veriler genellikle ilgili yazılıma ait özel formatlarında depolanmaktadır. Başka bir yazılıma geçişte ya da veri paylaşımında ek işlemler yapılması kaçınılmaz olmaktadır.

Paylaşım konusuna bir örnek ile yaklaşırsak: "A" CBS yazılımı kullanılan, "A" formatında, "A" veri yapısı ile depolanmış verilerin, "B" CBS yazılımında "B" veri yapısı ile kullanılması, işlenmesi nasıl olmaktadır? Bunun bir kereye özgü geçici bir işlem olmadığını, veri kalitesinin uygun olduğunu ve veri hacminin de küçük olmadığını da belirtmekte yarar vardır. Temelde iki yol söz konusudur; birinci yol "A" ve "B" sistemleri arasında format ve veri yapısı dönüşümüdür. İşlemler "A" tarafında yapılacak ise önce veri yapısı sonra format dönüşümü, "B" tarafında yapılacak ise de tam tersinin uygulanması mantıklıdır. Veri yapısı dönüşümü diğer sistem ile bütünleştirme olarak da düşünülebilir ve klasik yollarla "elle" ya da kodlanan bir arayüz ile sağlanabilir. Dönüşümler sırasında veri kayıplarının olabileceği, işlemlerin zaman alıcı ve yorucu olduğu bir çok kullanıcı tarafından bilinen durumlardır.

İkinci yol verinin "B" veri yapısında depolanmasını da doğrudan sağlayabilen GML vb bir format kullanmaktır. Araştırmanın konusu GML olduğundan makalede bu konunun üzerinde durulacak ve veri taşıyan GML dosyası GML dokümanı olarak anılacaktır. GML şemaları kullanılarak, veriler "A" yazılımında GML dokümanı haline dönüştürülür. Dokümanlar "B" yazılımı ile okunabilir, kullanılabilir. Veri yapısı, geliştirilecek bir yazılım ile "B" yapısına dönüştürülebilir.

2.1. GML Şemaları

GML dokümanlarının kullanılabilir olması, GML temel şemaları ile uyumlu olmasına bağlıdır. Uyum, XML dokümanının doğrulanabilir (validate) olması ile ifade edilebilir. Daha önce de belirtildiği gibi GML farklı sürümlerde yayımlanmıştır ve sürümlerde şemalar farklılıklar da göstermiştir. GML 1 sürümünde bileşen ve öznitelikleri tanımlamak için DTD (Document Type Descriptors-Doküman Tip Tanımlayıcıları) dokümanları kullanılmış, 2 ve 3. sürümlerde ise DTD yerine XML XSD (XML Schema Definition-XML Şema Tanımı) dosyalarının kullanımına geçilmiştir.

GML veri yapısı, verilerin tür, geometri ve topoloji bilgileri başta olmak üzere tanımlayıcı bileşenleri içermektedir (Şekil 2). GML 3. versiyonunda bileşenler 28 adet XSD şeması ile oluşturulmuştur.

Bir GML objesinin oluşturulabilmesi için gerekli en temel bileşen şemalarda aşağıdaki şekilde tanımlanmaktadır (OGC, 2004).

```
<element name="_GML" type="gml:AbstractGMLType"
abstract="true" substitutionGroup="gml:_Object"/>
<complexType name="AbstractGMLType" abstract="true">
<sequence>
<element ref="gml:metaDataProperty" minOccurs="0"
maxOccurs="unbounded"/>
<element ref="gml:description" minOccurs="0"/>
<element ref="gml:name" minOccurs="0"
maxOccurs="unbounded"/>
</sequence>
<attribute ref="gml:id" use="optional"/>
</complexType>
```

Bir nokta objesi GML uygulama şemasında GML temel şemalarına uygun olarak oluşturulmak istenirse, bileşenin tanımını aşağıdaki gibi olabilecektir.

```
<xs:element name="Nokta" type="NoktaTipi"
substitutionGroup="gml:_Feature"></xs:element>
<xs:complexType name="NoktaTipi">
<xs:complexContent>
<xs:extension
base="gml:AbstractFeatureType">
<xs:sequence>
<xs:element name="A"
type="xs:string"/>
<xs:element name="B"
type="xs:string"/>
<xs:element name="_Geometri"
type="gml:PointPropertyType"/>
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
```

Şema ile Nokta objesi global bir bileşen olarak tanımlanmıştır. Nokta objesinin bir GML vektör verisi olduğu "substitution Group" ile açıklanmıştır. Bu deyim GML temel şemalarında tanımlanan gml:_Feature objesinden türetildiğini göstermektedir. Devamında ise Nokta objesinin yapısını oluşturan NoktaTipi kompleks tipi tanımlanmıştır. Nokta verisi örnekte gml:AbstractFeatureType ifadesi ile temel GML objesinden doğrudan türetilmiştir. İzleyen satırlarda Nokta objesine atanacak olan öznitelikler tanımlanmıştır. A ve B isimli karakter türünden iki, Geometri isimli GML temel şemasında tanımlanan PointPropertyType (nokta türünden veriler için) türünden bir öznitelik tanımlanmıştır. "_Geometri" özniteliği, binanın X,Y ve varsa Z koordinat bilgilerini içerecektir. Bu uygulama şemasına ilişkin GML dokümanında Nokta verileri için örnek aşağıdaki gibi olabilir.

```
<Nokta gml:id="ID00001">
<A>Öznitelik A</A>
<B>Özniteli B</B>
<_Geometri><gml:Point
srsName="urn:opengis:def:crs:EPSG::28992">
<gml:pos
srsDimension="2">205891.031 445211.517</gml:pos>
</gml:Point></_Geometri >
</nokta>
```

Nokta gibi her GML objesinden oluşturulan yeni objenin bir tekil numarası (ID) bulunmaktadır. Bu sayede aynı obje türünden bir dokümanda ID'leri farklı olmak üzere birden fazla bulunabilmesi mümkün olmaktadır. Yine Nokta objesinde görüldüğü üzere şemada tanımlanan “_Geometri” özniteliği “gml:PointPropertyType” türünden türetildiği için şemadaki bileşende gösterilmemesine rağmen GML’de tanımlı türün özelliklerini taşıyarak Nokta’nın X, Y koordinat bilgilerini, koordinat sistemi (srsName) ve boyut (srsDimension) değerlerinin tanımlanmasına olanak sağlamıştır.

2.2. GML Örnekleri

Hollanda’nın TOP10NL formatı, dünyada ulusal standardizasyon kapsamında geliştirilmiş en önemli GML uygulama şemalarından birisidir. TOP10NL ile 1:5000 ile 1:25000 ölçekli verilerin depolanması, değişimi ve paylaşılması amaçlanmıştır (URL 2).

G-XML ise Japonya’da geliştirilen bir GML uygulama şemasıdır. G-XML, yine veri değişimi ve paylaşımı amaçlı olarak 1999’dan bu yana geliştirilmektedir. Bu proje; kamu, üniversiteler ve özel sektörün ortak çalışmalarıyla devlet destekli olarak yürütülmektedir (URL 3).

Diğer bir ulusal format oluşturma çalışması ise Almanya’daki “Normbasierte Austauschchnittstelle-Norm Tabanlı Veri Değişim Arayüzü”dür (NAS). NAS, AFIS-ALKIS-ATKIS projesi kapsamında geliştirilen bir GML uygulama şemasıdır. Asıl amaç veri değişimi için bir ulusal format geliştirilmesidir. Geliştirilen format GML tabanlıdır (URL 4).

Bir konuya yönelik olarak ulusal bir çalışma Kanada’daki CRML (Canadian Road Markup Language)’dir. CRML, Kanada’daki yol ağı verileri için tanımlanmış objeleri, özniteliklerini ve ilişkilerini içermektedir (URL 5).

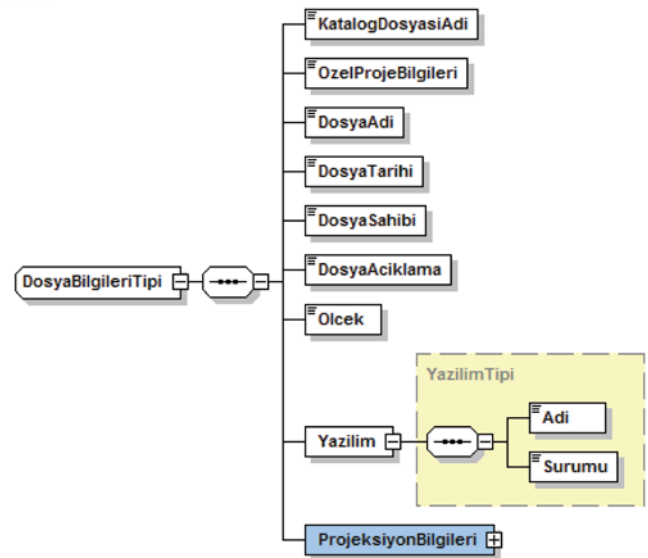
Ülkelerin ulusal çalışmalarının yanında çalışma grupları tarafından geliştirilen GML uygulama şemalarından biri CityGML’dir. CityGML, kentlerin üç boyutlu gösterimi için genel bilgi modelidir (URL 6). Formatın amacı, üç boyutlu kent modellerinin verilerinin depolanması ve değişiminin sağlanmasıdır.

CityGML benzeri diğer bir GML uygulama şeması GPML’dir (GPlates Markup Language). GPML, jeolojik bilgi modeli adı altında, jeolojik verilerin görüntülenmesi amacıyla geliştirilmiş bir formattır (URL 7). GeoSciML (URL 8) ve XMML (URL 9) benzer örnekler arasında sayılabilir.

Örneklerden de görüldüğü üzere GML tabanlı format oluşturma çalışmaları bir çok grup ve ülkede standardizasyon amacıyla devam etmektedir. Bu sayede ulusal ya da çalışma grupları anlamında standartlaştırılmış bir veri taşıma ya da işleme formatı oluşturulmuş olmakta, aynı zamanda bu format uluslararası GML standardına da uymaktadır.

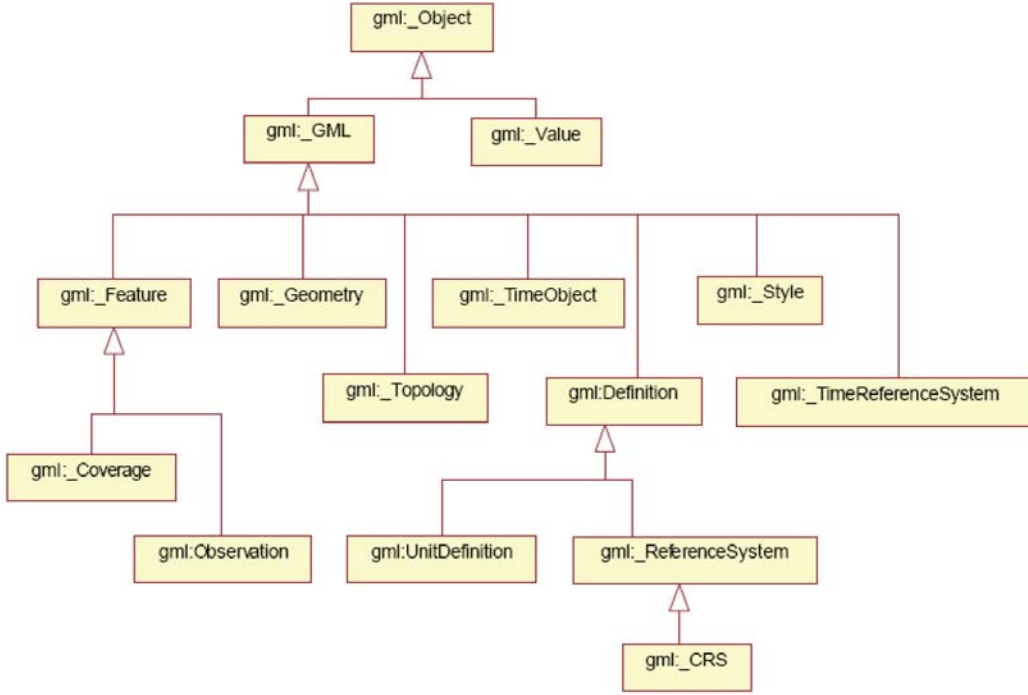
3. UVDF

Coğrafi verinin değişiminin ve paylaşımının dünyada giderek artan önemi ülkemizde de bu konuda çalışmalar yapılmasına yol açmıştır. Veri değişimi açısından ülkemizdeki en önemli ve somut adım, bir veri değişim formatı olan UVDF’nin BÖHHBÜY eki olarak yayımlanması ile atılmıştır. UVDF çalışmaları, 1988 yılında yürürlüğe giren Harita Yapım Yönetmeliğinin güncellenmesi ve BÖHHBÜY oluşturulması süreciyle başlamıştır (Çelik, 2007). UVDF, XML tabanlıdır ve iki ayrı XML şeması ile tanımlanmıştır. İlk şema objelerin grafik içeriklerinin oluşturulduğu geometri şemasıdır. Bu şemada öncelikle her dosyada bulunması gerekli olan meta veriler tutulmaktadır (Şekil 1).

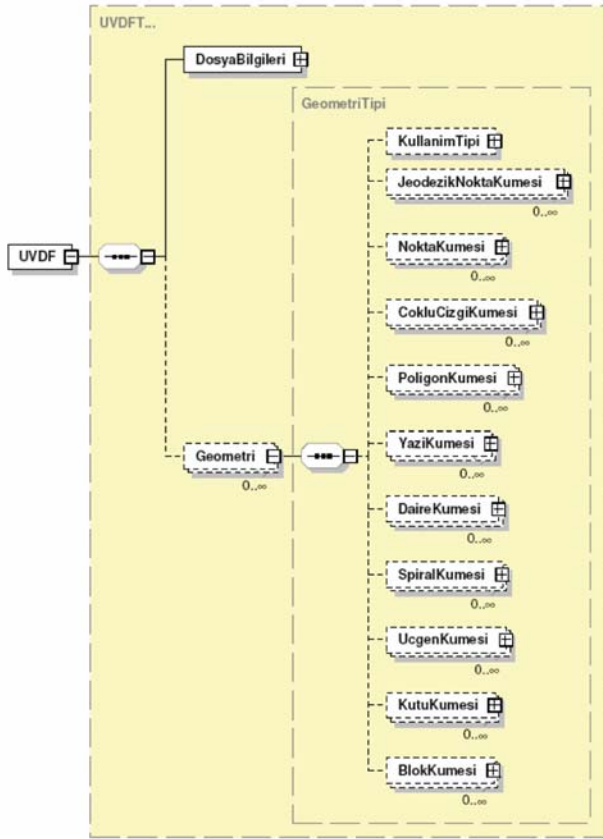


Şekil 1: UVDF dosya bilgileri bileşeni

Geometri şemasında çoklu çizgi, nokta, poligon ve diğer kümeler tanımlanmış, içlerine yol, bina vb detaylar yerleştirilmiştir (Şekil 3). BÖHHBÜY’ün eki olan DÖKK ile tanımlanan detay türleri obje bazında tanımlanmamış, bunun yerine kümeler içinde tanımlanmıştır (TKGM,2007). Oluşturulan detayların altında VTESTipi (Veri Tabanı Erişim ve Sembol Kodu tipi) adlı kompleks tip, XML bileşeni olarak tanımlanmıştır. Bu bileşen bir objenin DÖKK’a göre hangi objeyi temsil ettiği (sınıfı), kimlik numarası (ID) ve geometri sembol kodu (kod) özniteliklerini taşımaktadır. Bir bina objesi, PoligonKümesi bileşeninin sınıf özniteliğinin değeri “Yapı” olan bir poligon alt bileşeni olarak tutulacaktır. Şekil 4’de görüldüğü üzere bir “Yapı” objesi geometrik özelliklerinin yanısıra ID ve Kod özniteliklerini ve değerlerini taşımaktadır.



Şekil 2: GML obje düzeni (OGC, 2004)



Şekil 3: UVDF Geometri şeması obje tipleri

UVDF'nin ikinci XML şeması DÖKK'te yer alan öznitelikleri içermektedir. Öznitelikler UVDF dokümanlarında "... hangi projede ne gibi sözel bilgi tutulacağı önceden tasarlanamayacak kadar geniş bir konudur. Bu açıdan, bir UVDF dosyasında projeye özgü tablo verileri varsa o proje ismi (UVDF-Takbis, UVDF-Orman, UVDF-Su, UVDF-Elektrik

vb.) ile anılmalıdır" şeklinde ifade edilmiştir. Uygulamada sözel veriler mevcut ise bu veriler ikinci bir XML dosyası içinde, her bir detayla ID ile ilişkilendirilmiş olarak tutulacaktır. Tablo dosyasında DÖKK ile tanımlanan öznitelikler ile projeye özel olarak bulunabilecek diğer öznitelikler de yer alabilmektedir.

4. Ulusal GML Uygulama Şeması Tasarımı (UVDF-GML)

Araştırmada, BÖHHBÜY DÖKK tanımlamaları esas alınarak UVDF-GML oluşturulmuştur. Ayrıca mevcut UVDF şemalarında bulunan özellikler de mümkün olduğu ölçüde korunmuştur.

GML ülkemizde bugüne kadar akademik çalışmalar dışında çok fazla kullanılmamıştır. Bu çalışma ile UVDF uluslararası standartlara uygun, modern CBS ve UKVA ihtiyaçlarını karşılayabilecek bir yapıya getirilmeye çalışılmıştır. Çalışmada, Mayıs 2005'te yayımlanan GML 3.1.1 versiyonu kullanılmıştır. Bunun nedeni günümüzde bir çok yazılımın 3.1 versiyonunu desteklemesidir. Uygulama şemasının son versiyon olan 3.2.1'e uyumluluğu kolayca sağlanabilmektedir. UVDF-GML, iki uygulama şeması ile hazırlanmıştır. Bunlardan biri DÖKK geometrik objelerinin GML tanımlarının yer aldığı geometri şemasıdır. Diğer ise DÖKK özniteliklerinin ve ön değerlerinin tanımlandığı ÖznitelikDetay şemasıdır. Öznitelik Detay şemasında, DÖKK da yer alan tüm öznitelikler ve ön değerleri birer simpleType XML şema bileşeni olarak oluşturulmuştur. Örneğin DÖKK'te "KOT_DEGERI" adıyla belirlenen öznitelik bileşeni şemada Şekil 5'de gösterildiği gibi hazırlanmıştır. Aynı öznitelik XML kodlarıyla aşağıdaki gibidir.

Sınıf	ID	Kod	Alan
1 YAPI	2	1202	Alan
2 YAPI	1	1201	Alan

Y	X	Z
1	704925.238000	4229177.965600
2	704949.219600	4229178.218000
3	704949.219600	4229157.517600
4	704925.490400	4229158.780000

Şekil 4: UVDF dosyasında yapı objesinin oluşturulması

```

-<xs:simpleType name="KOT_DEGERI">
  <xs:annotation>
    <xs:documentation>DETAYIN KOT DEĞERİ (Z DEĞERİ)
(KODLU)</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:integer">
    <xs:enumeration value="1">
      <xs:annotation>
        <xs:documentation>GEOMETRİK NİVELMAN
KOTU</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="2">
      <xs:annotation>
        <xs:documentation>TRİGONOMETRİK NİVELMAN
(PREZİSYONLU)</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
  </xs:restriction>
</xs:simpleType>

```

Her öznitelik değeri (enumeration), bunların açıklamaları (documentation) ve öznitelik değerinin tipi belirlenmiş ve kullanıcının bu değerler dışında giriş yapması engellenmiştir. Örneğin KOT_DEGERI=1 olan bir detayın öznitelik değeri “Geometrik Nivelman Kotu” na karşılık gelmektedir. Bir kullanıcı ya da geliştirici eğer farklı bir değer girecek olursa dosya şema ile uyumlu olmayacaktır. Şekil 5’de görülen ÖznitelikDetay şemasında toplam 60 adet öznitelik ve ön değeri XML şema bileşeni şeklinde tanımlanmıştır.

Geometri şemasında ise DÖKK Detay Tablosunda yer alan geometrik detaylar ve bunların öznitelik değerleri ile varsa ön değerler; veri tipleri ve sınırlamaları kullanılarak birer GML objesi şeklinde tanımlanmıştır. UVDF-GML’deki yaklaşım UVDF’deki yaklaşımdan farklı olarak her bir detay bir obje olacak şekilde objeye-yönelik sistem kurallarına uygun olarak oluşturulmuştur. Bu durum UVDF’de her bir detayın birer öznitelik değeri ile belirlenmesi şeklinde yapılmıştır. Şemaların içinde genel XML dışında, üç adet isim

uzayı kullanılmıştır. Bunlar GML’den gelen “gml”, Öznitelik Detay’ dan gelen “ozdt” ve geometri şemasının kendi isim uzayı “uvdf” dir.

Şemada objeler UVDF kök bileşeni altında tanımlanmıştır. UVDF kök bileşeni UVDF’de yer alan DosyaBilgileri bileşeni ve içeriği aynen korunmuştur (Şekil 1). Bunlar dosya ve içeriği hakkında özet bilgi veren bileşenlerdir.

DÖKK’deki her bir geometrik detay, birer global bileşen ve bu bileşenin global tip bileşeni olmak üzere iki adımda tanımlanmıştır. Bu şekilde şemanın daha düzenli olması ve tip bileşenlerinin türemiş farklı objeler ve detay gruplaması (FeatureCollection) için tekrar kullanılabilirliği sağlanmıştır. Örneğin DÖKK’de yer alan Parsel objesi;

```

<xs:element name="Parsel" type="uvdf:ParselTipi"
substitutionGroup="gml:_Feature">
  <xs:annotation>
    <xs:documentation>İLK TESİS KADASTROSU
VEYA İMAR UYGULAMASI İLE OLUŞAN KADASTRO VEYA
İMAR ADASI İÇERİSİNDE KALAN MÜLKİYETİ TESCİLLİ EN
KÜÇÜK BİRİM ALAN</xs:documentation>
  </xs:annotation>
</xs:element>

```

şeklinde oluşturulmuştur. Global bir Parsel bileşeni tanımlanmış, bunun tip özniteliği değeri ParselTipi verilmiştir. Parsel bileşeninin “substitutionGroup” olarak tanımlanan “gml:_Feature” ifadesiyle, bu bileşenin herhangi bir geometrik GML objesinin özelliklerini taşıyacağı gösterilmektedir. “_Feature” değişim grubunun tanımlanması ile bu objenin GML dosyasında “featureMember” bileşeni altında gruplandırılması sağlanmaktadır. Parselin içeriğini aldığı tip global bileşeni ise “ParselTipi” adı ile “complexType” türünden oluşturulmuştur. İçeriği karmaşık tipte ve GML şemalarından “AbstractFeatureType” dan türetilmiştir. Bu sayede ParselTipinin bir detay olduğu ve GML okuyan bir yazılım tarafından bulunabilir olması sağlanmaktadır.

xs:simpleType	
name	KOT_DEGERI
xs:annotation	xs:documentation DETAYIN KOT DEĞERİ (Z DEĞERİ)(KODLU)
xs:restriction	base xs:integer
xs:enumeration	value 1
xs:annotation	xs:documentation GEOMETRİK NİVELMAN KOTU
xs:enumeration	value 2
xs:annotation	xs:documentation TRİGONOMETRİK NİVELMAN (PREZİSYONLU)
xs:enumeration	value 3
xs:annotation	xs:documentation TRİGONOMETRİK NİVELMAN (KONVANSİYONEL)
xs:enumeration	value=4
xs:enumeration	value=5

Şekil 5: KOT_DEGERI özniteliği ve alacağı değerlerin şemada oluşturulması (tablosal gösterim)

```

<xs:complexType name="ParselTipi">
  <xs:complexContent>
    <xs:extension base="gml:AbstractFeatureType">
      <xs:sequence>
        <xs:element name="DetaySinifKodu"
type="xs:string" fixed="MA"/>
        <xs:element name="DetayKodu"
type="xs:string" fixed="0002"/>
        <xs:element name="ADA_NUMARASI" type="xs:string"/>
        <xs:element name="ADI_NUMARASI" type="xs:string"/>
        <xs:element name="ITIRAZ_DURUMU"
type="ozdt:ITIRAZ_DURUMU"/>
        <xs:element name="ITIRAZ_NEDENI"
type="ozdt:ITIRAZ_NEDENI"/>
        <xs:element name="MALIK" type="xs:string"/>
        <xs:element name="NITELIK_01"
type="ozdt:NITELIK_01"/>
        <xs:element name="NOKTA_SAYISI" type="xs:short"/>
        <xs:element name="OLCU_HESAP_FARK"
type="gml:MeasureOrNullListType"/>
        <xs:element name="OLCU_KROKISI" type="xs:short"/>
        <xs:element name="OLUSUMU" type="ozdt:OLUSUMU"/>
        <xs:element name="TAPU_YUZOLCUMU"
type="gml:AreaType"/>
        <xs:element name="TECVIZ_DURUMU"
type="ozdt:TECVIZ_DURUMU"/>
        <xs:element name="TECVIZ_SINIRI" type="xs:short"/>
        <xs:element name="_Geometri"
type="gml:SurfacePropertyType"/>
        <xs:element ref="uvdf:_genelOznitelik" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

Daha sonraki bölümde tipin DÖKK'te belirlenen öznitelik değerleri; veri tiplerine göre tanımlanmıştır. Bu öznitelikler bulunması ve değer girilmesi zorunlu öznitelik değerleri şeklinde yer almıştır. DetaySinifı ve DetayKodu her detayda bulunan öznitelikler olup, bunlar DÖKK'te hiyerarşik olarak ait oldukları detay sınıfını ve bu sınıf altındaki detay kod numarasını göstermektedir. Bu değerler şemada her obje için sabit öznitelikler şeklinde tanımlanmıştır.

ITIRAZ_DURUMU gibi ön değer sahip özniteliklerin özellikleri ise daha önce oluşturulan ÖznitelikDetay şemasından alınmaktadır. Bu değerler sadece ÖznitelikDeğer şemasında tanımlanan değerler olabilmektedir. Detayın geometrik özelliklerini taşıyan öznitelik grubu ise her detay için “_Geometri” altında toplanmış ve GML geometri şemalarında belirlenmiş geometrik şekil tanımlamalarından detay özelliklerine uyan bir tip seçilmiştir. Parsel için kapalı alan türüne uygun olan “SurfacePropertyType” seçilmiştir.

Birinci Bölümde de belirtildiği gibi, DÖKK'de bir detay için zorunlu öznitelikler belirlenmiştir. Projenin amacına göre her bir detay türü farklı öznitelikler içerebilmektedir. Bu durumun çözümü ise UVDF-GML'de “_genelOznitelik” bileşeni ile gerçekleştirilmiştir. Yani, bir UVDF-GML detayı projeye göre istenilen sayıda farklı öznitelik taşıyabilmektedir. Bunun için genel veri türlerini kapsayacak, cinsine özgü bildirim yapabilecek (generic) bileşenler oluşturulmuştur. Bu bileşenler tarih, yazı, tam sayı ve ondalıklı sayı şeklinde belirlenmiştir. UVDF-GML dosyası oluşturulurken kullanıcının “_genelOznitelik” soyut bileşeninden oluşturulan bileşeni kullanması gerekmektedir. XML bileşeni özniteliğinin veri tipini, “name” adını ve “value” bileşeni de özniteliğinin değerini

taşıyacaktır. Böylece tüm detay ve özneliklerinin farklı dosyalar yerine tek bir dosyada taşınması sağlanmıştır.

```
<xs:element name="_genelOznitelik"
type="uvdf:_GenericOznitelikType" abstract="true"/>
<xs:complexType name="_GenericOznitelikType" abstract="true">
  <xs:attribute name="name" type="xs:string use="required"/>
</xs:complexType>

<xs:element name="intOznitelik" type="uvdf:IntOznitelikType"
substitutionGroup="uvdf:_genelOznitelik"/>

<xs:complexType name="IntOznitelikType">
  <xs:complexContent>
    <xs:extension
base="uvdf:_GenericOznitelikType">
      <xs:sequence>
        <xs:element name="value"
type="xs:integer" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

BÖHHBÜY DÖKK'te yer alan 146 adet detay türü UVDF-GML objesi olarak geometri şemasında tanımlanmıştır. Bunların yanısıra CBS'de yoğun olarak kullanılan Grid, TIN gibi farklı veri türlerini GML şemalarını kullanarak UVDF-GML dosyasında oluşturmak ve taşımak mümkündür. UVDF-GML şemasında bu iki veri türü de aşağıdaki şekilde tanımlanmıştır.

```
<xs:element name="TIN" type="uvdf:TinType"
substitutionGroup="gml:_Feature"/>
<xs:complexType name="TinType">
  <xs:complexContent>
    <xs:extension
base="gml:AbstractFeatureType">
      <xs:sequence>
        <xs:element name="_Geometri"
type="gml:TinType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="Grid" type="uvdf:GridType"
substitutionGroup="gml:_Feature"/>
<xs:complexType name="GridType">
  <xs:complexContent>
    <xs:extension
base="gml:AbstractFeatureType">
      <xs:sequence>
        <xs:element
name="_Geometri" type="gml:RectifiedGridType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

BÖHHBÜY DÖKK'te bulunmayan ancak ek olarak UVDF-GML şemasında yer alan başka bir obje türü yol orta çizgileridir. Yol orta çizgisi verilerinin UVDF-GML dosyasında taşınabilmesi için aşağıdaki yapı oluşturulmuştur.

```
<xs:complexType name="YolOrtaCizgisiTipi">
  <xs:complexContent>
    <xs:extension
base="gml:AbstractFeatureType">
      <xs:sequence>
        <xs:element name="DetaySinifKodu"
type="xs:string" fixed="KY"/>
        <xs:element name="DetayKodu"
type="xs:string" fixed="0020"/>
        <xs:element name="_Geometri"
type="gml:CurvePropertyType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Detayların, DÖKK hiyerarşik yapısında tutulması için detay koleksiyon grupları oluşturulmuştur. Böylece her bir detay UVDF-GML içinde ayrıca tutulabileceği gibi detay sınıfları içinde de gruplar halinde tutulabilecektir. Bu amaçla DÖKK'da yer alan detay sınıflarına ait koleksiyon grup bileşenleri oluşturularak, yer alacak detaylar bu koleksiyonlar altında tanımlanmıştır. Örneğin detay sınıf kodu "IA" olan İdari Alan/Sınır/Merkez detay sınıfı "İdariAlanSinifi" detay koleksiyon bileşeni altında

```
<xs:element name="IdariAlanSinifi" type="uvdf:IdariAlanSinifTipi"
substitutionGroup="gml:_FeatureCollection"/>
<xs:complexType name="IdariAlanSinifTipi">
  <xs:complexContent>
    <xs:extension
base="gml:AbstractFeatureCollectionType">
      <xs:sequence>
        <xs:element name="IdariAlan"
type="uvdf:IdariAlanTipi" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="IdariSinir"
type="uvdf:IdariSinirTipi" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="IdariMerkez"
type="uvdf:IdariMerkezTipi" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="SorumlulukAlani"
type="uvdf:SorumlulukAlaniTipi" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

şeklinde tanımlanmış, böylece DÖKK hiyerarşisi içinde veri saklamak mümkün olmuştur.

4.1. UVDF-GML Şemalarının Uygulanması

UVDF-GML Uygulama şeması ile bazı örnek verilerin depolandığı bir GML dokümanı hazırlanmış ve GML şemalarını okuyabildiği belirtilen yazılımlardan bazıları kullanılarak testler yapılmıştır. Testin amacı herhangi bir eklenti geliştirilmeden bu yazılımların BÖHHBÜY'e uygun UVDF-GML'i görüntüleyebilme durumunun sınanmasıdır.

Testlerde, ArcGIS 9.2³, ücretsiz olarak kullanılabilen Tatuk Viewer⁴, Snowflake Software GML Viewer⁵ ve Gaia 3.0⁶ yazılımları kullanılmıştır. Örnek UVDF-GML dosyası elle hazırlanmış ve tüm detaylar yerine farklı tür verilerden en çok karşılaşılabilecek olan detaylar seçilmiştir. Bu detay türleri; tek ağaç (BR0005), parsel (MA0002), karayolu (KY0002), işyeri (YA0004), eş yükseklik eğrisi (AR0010), çeşme (HD0002), yol orta çizgisi (KY0020), yapı yerleşim alanı (YA0001), idari alan (IA0001), idari mesken (IA0003) ve ada (MA0001) olmak üzere nokta, çizgi ve alan veri türlerini temsil edecek şekilde belirlenmiştir. Detaylar UVDF-GML uygulama şemasına uygun olarak özniteliklerine değerler verilerek, örneğin Ada detayı;

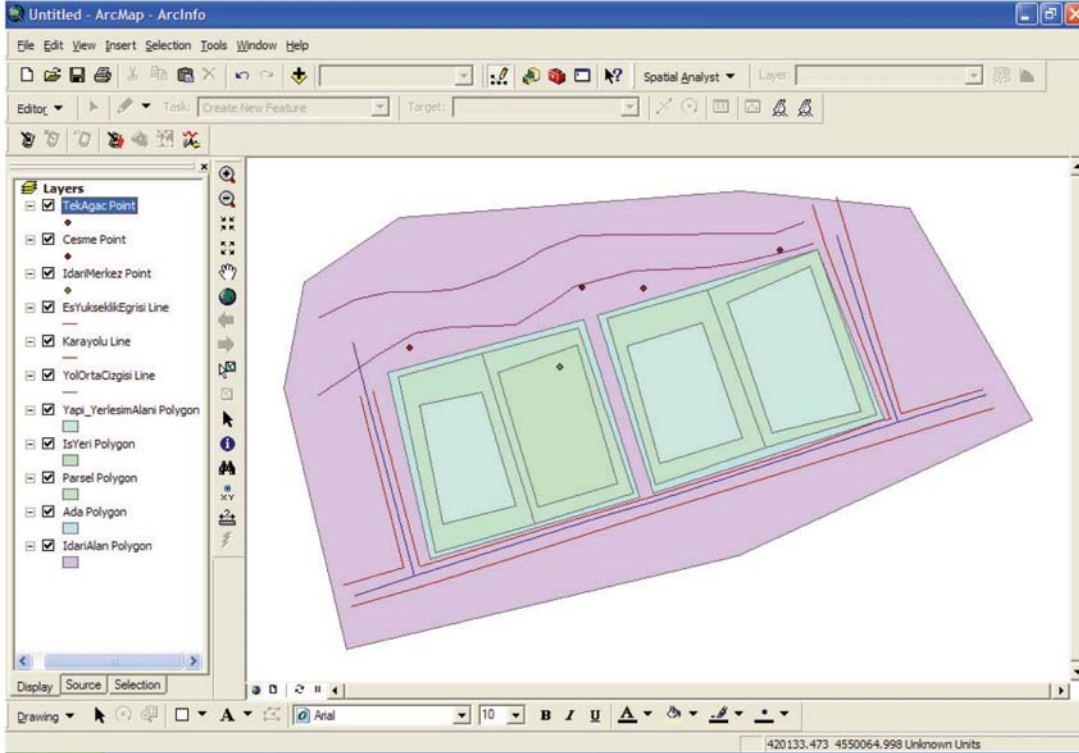
```
<gml:featureMember>
  <uvdf:Ada gml:id="ID986">
    <uvdf:DetaySinifKodu>MA</uvdf:DetaySinifKodu>
    <uvdf:DetayKodu>0001</uvdf:DetayKodu>
    <uvdf:ADI_NUMARASI>23</uvdf:ADI_NUMARASI>
    <uvdf:TARİH>2001-09-09</uvdf:TARİH>
    <uvdf:_Geometri>
      <gml:Polygon>
        <gml:exterior>
          <gml:LinearRing>
            <gml:posList>420148.112487793 4550073.56567383
420173.986694336 4550080.51068115 420181.340515137
4550057.22387695 420153.695922852 4550049.05310059
420148.112487793 4550073.56567383</gml:posList>
          </gml:LinearRing>
        </gml:exterior>
      </gml:Polygon>
    </uvdf:_Geometri>
  </uvdf:Ada>
</gml:featureMember>
```

şeklinde featureMember bileşeni altında kodlanmıştır. Hazırlanan UVDF-GML dosyası yazılımlar ile okunabilmiş ve içerdiği veriler ekrana grafik olarak sorunsuzca çizdirilebilmiştir (Şekil 6, 7, 8, 9). Yazılımlar ile featureMember bileşeni altında ayrı ayrı tanımlanan detaylar okunabilmiştir. Ancak, bazı yazılımlar ile featureMembers bileşeni altında bulunan değişik detayları okumada ve görüntülemeye sorunlar yaşanmıştır. Yazılımlar ile detayların özniteliklerine göre sorgulama ve basit analizler başarıyla yapılmıştır. Ancak, bağımsız tanımlanan genelÖznitelik bileşenin sorgulanması bazı yazılımlar ile yapılamamıştır.

5. Sonuç

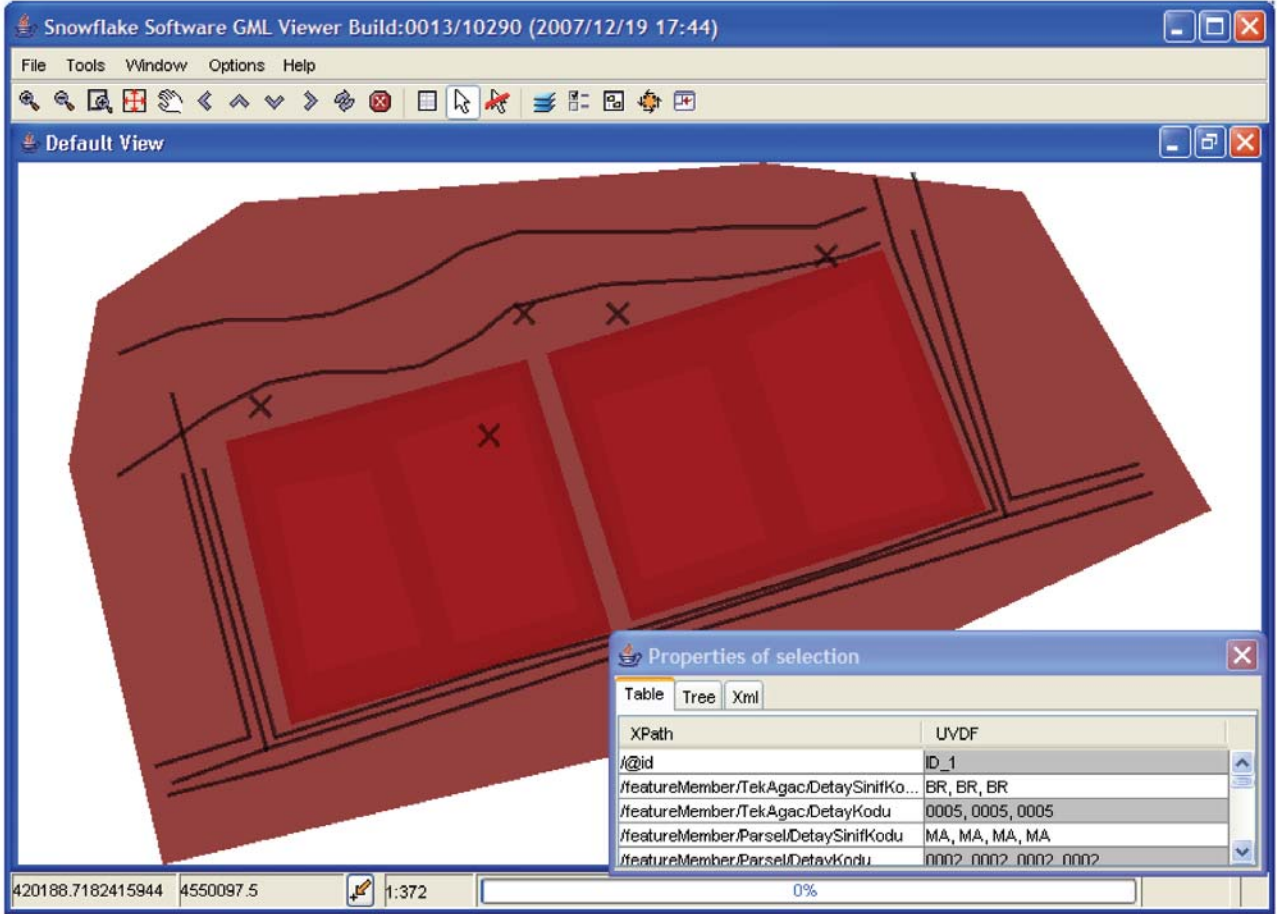
CBS ve UKVA göz önünde bulundurulduğunda, UVDF'nin GML gibi kapsamlı ve uluslararası bir dayanağı olan yapıya sahip olması veri paylaşımı ve birlikte işlerlik açısından büyük kolaylıklar ve düzen sağlayacaktır. GML bünyesindeki objeye yönelik yaklaşım, ilişkiler, topoloji, meta veri ve zengin detay türleri açısından dünyada kabul görmüş ve uygulanır bir format haline gelmiştir.

Çalışmada gerçekleştirilen örnek UVDF-GML, ülkemiz standartlarının uluslararası standartlara, belirtilere uygun olarak yönlendirilebileceğini ortaya koymaktadır. Geliştirilen UVDF-GML dosyaları bir çok yazılım tarafından eklentiye ihtiyaç duyulmadan okunabilmektedir. Ancak, mevcut verilerin bu şemalara uygun dosyalara dönüştürülmesi için yazılımlara eklentilerin geliştirilmesi gerekmektedir. Böylece kullanıcıların başka bir araca gereksinim duymadan verileri görüntüleyerek kullanmaları, başka formatlar dönüştürmeleri mümkün olmaktadır.

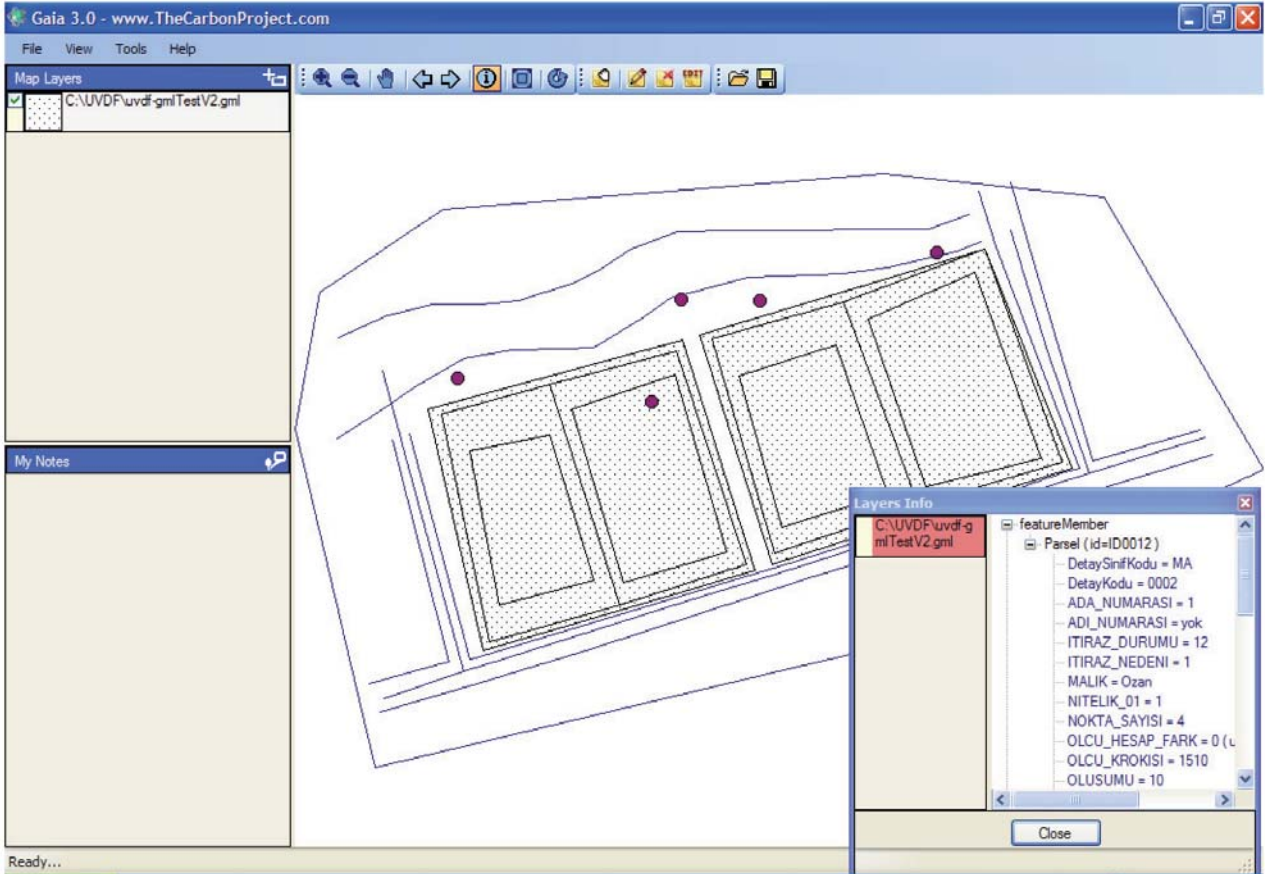


Şekil 6: Test verisinin ArcGIS 9.2 yazılımı ile görüntülenmesi

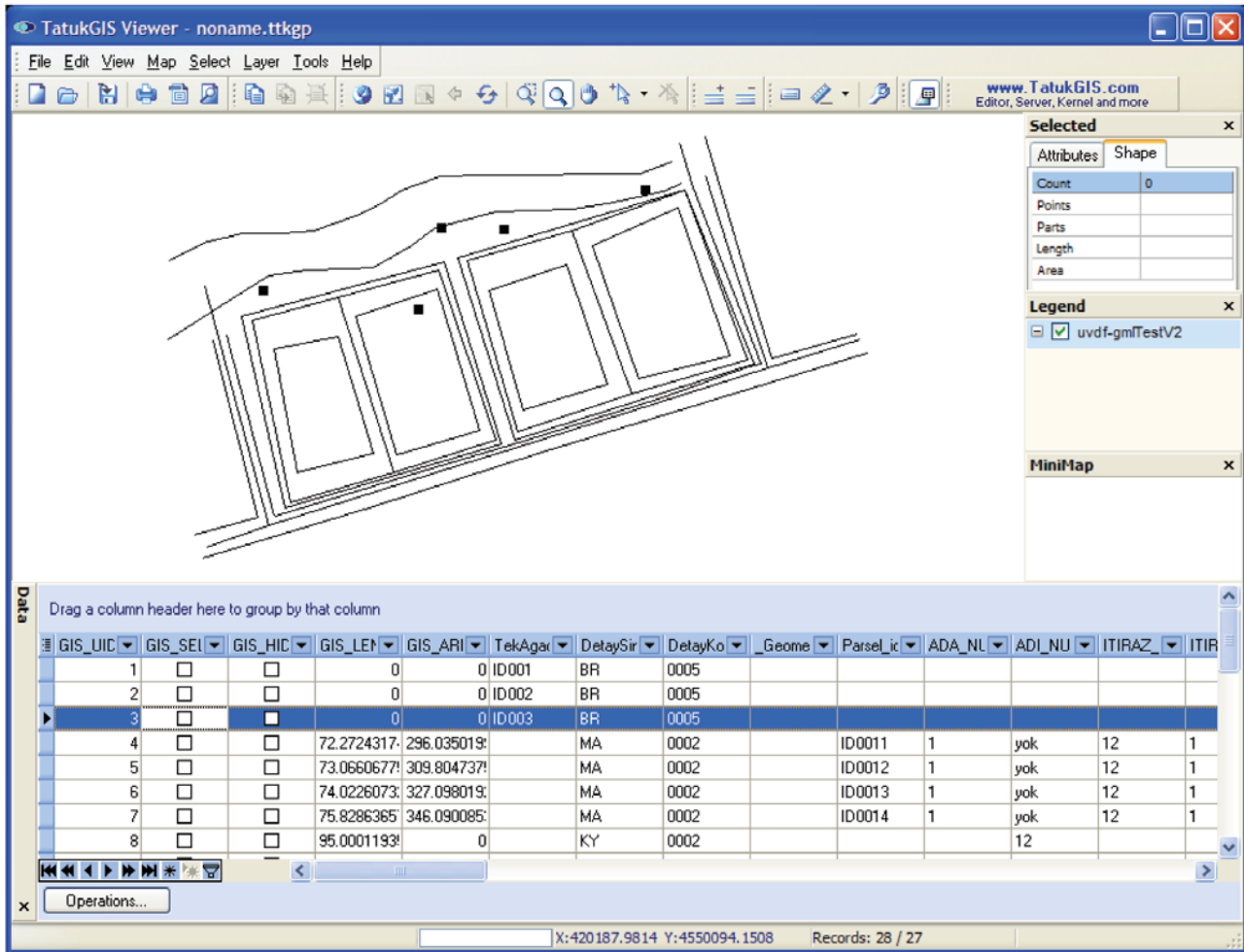
³www.esri.com, ⁴www.tatukgis.com, ⁵www.snowflakesoftware.co.uk, ⁶www.thecarbonproject.com/gaia.php



Şekil 7: Test verisinin Snowflake yazılımı ile görüntülenmesi



Şekil 8: Test verisinin Gaia 3.0 yazılımı ile görüntülenmesi



Şekil 9: Test verisinin TatukGIS yazılımı ile görüntülenmesi

Kaynaklar

- BATUK F., ÖZTÜRK D., EMEM O: **Türkiye Ulusal Mekansal Veri Altyapısı İçin Temel Veriler**, Jeodezi, Jeoinformasyon ve Arazi Yönetimi, Harita ve Kadastro Mühendisleri Odası Yayın Organı, 2007/1 Sayı 96, 3-12, Ankara
- CHANG-TİEN LU, RAİMUNDO F. DOS SANTOS JR, LAKSHMİ N. SRİPADA, YUFENG KOU: **Advances in GML for Geospatial Applications**, Geoinformatica 2007, 131-157, Springer Hollanda
- EMEM O: **Mekansal Veri ve Bilgi Altyapısının Uygulamalı Olarak Geliştirilmesi**, Doktora Tez Çalışması, YTU, Fen Bilimleri Enstitüsü., 2007
- OGC: **OpenGIS® Geography Markup Language (GML) Implementation Specification, Versiyon 3.1.0**, OpenGIS Consortium Inc. , 2004
- TKGM, www.tkgm.gov.tr/turkce/dosyalar/diger/uvdf/anadokuman.doc, 2007
- URL 1: <http://www.w3.org/XML/>, W3C, eXtensible Markup Language, XML, Eylül 2008

- URL 2: <http://www.kadaster.nl>, Kadaster, Eylül 2008
- URL 3: <http://www.dpc.jipdec.or.jp/gxml/contents-e/>, G-XML, G-XML Project, 11.01.2009
- URL 4: <http://www.adv-online.de/exteng/broker.jsp?uMen=01a700d3-6ed6-0bfb-8f23-50376a112976>, ADV, AdV projects AFIS, ALKIS and ATKIS, 11.01.2009
- URL 5: <http://www.statcan.ca/bsolc/english/bsolc?catno=92-500-X>, Statcan, Road Network File, 11.01.2009
- URL 6: <http://www.citygml.org/>, CityGML, 11.01.2009
- URL 7: <http://www.earthbyte.org/Resources/GPGIM>, GPML, GPlates Geological Information Model: Resource Page, GPGIM, 11.01.2009
- URL8: <https://www.seegrid.csiro.au/twiki/bin/view/CGIModel/GeoSciML>, GeoSciML, GeoSciML - Geological SciencesML, 11.01.2009
- URL9: <https://www.seegrid.csiro.au/twiki/bin/view/Xmml/WebHome>, GMML, XMML-eXploration and Mining Markup Language, 11.01.2009