

TKGM CityGML Veri Modelinin 3B WebCBS Uygulamaları Açısından İrdelenmesi

Investigation of the TKGM CityGML Data Model in the Context of 3D WebGIS Applications

Ziya Usta^{1*}, Alper Tunga Akın², Çetin Cömert²

¹Artvin Çoruh Üniversitesi, Mühendislik Fakültesi, Harita Mühendisliği Bölümü, 08100, Artvin/Türkiye.

²Karadeniz Teknik Üniversitesi, Mühendislik Fakültesi, Harita Mühendisliği Bölümü, 61080, Trabzon/Türkiye.

ARAŞTIRMA MAKALESİ

*Sorumlu yazar:

Ziya Usta
ziyausta@artvin.edu.tr

doi: 10.48123/rsgis.1059389

Yayın süreci

Geliş tarihi: 18.01.2022
Kabul tarihi: 01.03.2022
Basım tarihi: 14.03.2022

Özet

CityGML, kent modellerinin depolanması ve transfer edilmesi için tasarlanmış, açık bir veri modeli ve veri formatıdır. CityGML, 3B CBS uygulamalarında en çok kullanılan veri modeli ve veri formatıdır. TKGM tarafından gerçekleştirilen 3B Kent Modelleri ve 3B Kadastro Projesi kapsamında da CityGML veri modeli ve formatı kullanılmaktadır. Bu proje kapsamında, CityGML veri modeli, veri modelinde tanımlı olmayan 3B kadastro çalışmalarında gerekli verilerin tutulması için genişletilmelidir. Bu bağlamda TKGM, örnek "TKGM CityGML" adlı bir veri modeli tasarlayarak, geliştiriciler için kullanılmak üzere bu veri modeline uygun olarak hazırlanmış örnek bir dosya oluşturmuştur. Bu çalışmada TKGM CityGML veri modeli ve örnek CityGML dosyası incelenmiştir. Böylece TKGM CityGML veri modeline göre gerçekleştirilecek 3B CBS projelerinde veri modeli ve veri formatından dolayı oluşabilecek sorunlar irdelenmiştir. Bu veri modelinin, veri tekrarına sebep olduğu, içerisindeki verilerin ayrıştırılmasının zor olduğu görülmüştür. Özellikle web uygulamaları için TKGM CityGML veri modeli ve formatının uygun olmadığı görülmüş, bu çalışma kapsamında tespit edilen sorunları gideren yeni bir veri modeli ve formatı önerilmiştir.

Anahtar kelimeler: 3B CBS, 3B WebCBS, CityGML, TKGM CityGML, CityJSON

Abstract

CityGML is designed as an open data model and XML-based format for the storage and exchange of 3DCMs. CityGML is the most used data model and data format for 3D GIS applications. It is also used by nationwide 3D cadastre project called 3D City Models and 3D Cadastre Project by General Directorate of Land Registry and Cadastre. In the context of this project, CityGML data model should be extended to store city objects that are not defined in the existing CityGML data model. For this purpose, General Directorate of Land Registry and Cadastre designed a new data model called "TKGM CityGML" and prepared a sample CityGML file for TKGM CityGML data model to be used by developers. In this work, TKGM data model and sample CityGML file has been investigated. The problems that may arise in the 3D applications that use TKGM CityGML data model have been determined. This work indicates that this data model causes data duplication, and this data format is not easy to parse, just like the data model itself. Particularly, this data model and format was not suitable for web applications and in this study a new data model and format have been suggested to eliminate aforementioned problems.

Keywords: 3D GIS, 3D WebGIS, CityGML, TKGM CityGML, CityJSON

1. Giriş

3B Kent Modeli (3BKM), başta binalar olmak üzere, kent objelerinin 3B geometrilerinin ve özniteliklerinin dijital bir temsildir. 3BKM'leri 3B temsiline gerektiği, kentsel altyapı yönetimi, navigasyon, simülasyon, planlama, acil durum yönetimi, kent planlama, turizm, taşınmaz değerlendirme gibi pek çok alanda, silüet analizi, bina cephelerinin güneş potansiyellerinin belirlenmesi, kentteki gürültü dağılımının tespiti gibi 3B geometrik temsil gerektiren analizlerde kullanılmaktadır. 3BKMlerin kullanım alanları ile ilgili daha detaylı bir literatür taraması (Biljecki vd. 2015b)'te bulunmaktadır.

Konunun daha iyi anlaşılması bakımından 3B ve 2.5B temsil farkına değinmek gerekir. Konumsal bilgi işlemede pek çok uygulamanın temelini oluşturan Sayısal Yükseklik Modelleri (SYM) 2.5B bir temsile işaret eder. SYM, topoğrafik yüzeyin bir fonksiyon (f) ile temsil edilerek, matematik olarak tanımlı hale getirilmesi ilkesine dayanır. "f" fonksiyonunun belirlenmesi bir "yüzey uydurma" problemidir. Matematik olarak tanımlı bir yüzey ile artık bu tanımın elverdiği bir çok uygulama ya da analiz yapılabilir duruma gelir; Ancak bu tür bir tanımlamada, herhangi bir noktanın yüksekliği, $z = f(x,y)$ bağıntısı ile hesaplanacağından, aynı noktaya ait tek bir z değeri elde edilebilir. Böyle bir tanımlama ya da "temsil" 2.5B olarak adlandırılır. 3B temsilde ise aynı noktaya ait birden çok "z" değeri temsil edilebilir. Diğer bir anlatımla, 3B temsil 2.5B'nin aksine yalnızca "yüzey" in değil, yüzeyin altı ve/veya üstünün de tanımlanabildiği bir temsildir. Dolayısıyla 3B temsil gerektiren analizlerde, SYM temsili yetersiz kalacaktır. Çeşitli amaçlı görülebilirlik analizleri, hava akımı analizleri, gölge analizleri, çeşitli jeolojik analizler gibi yüzey üstü ve altında tanımlı hale getirilmesi gereken nesnelere içeren analizler 3B temsil gerektirecektir.

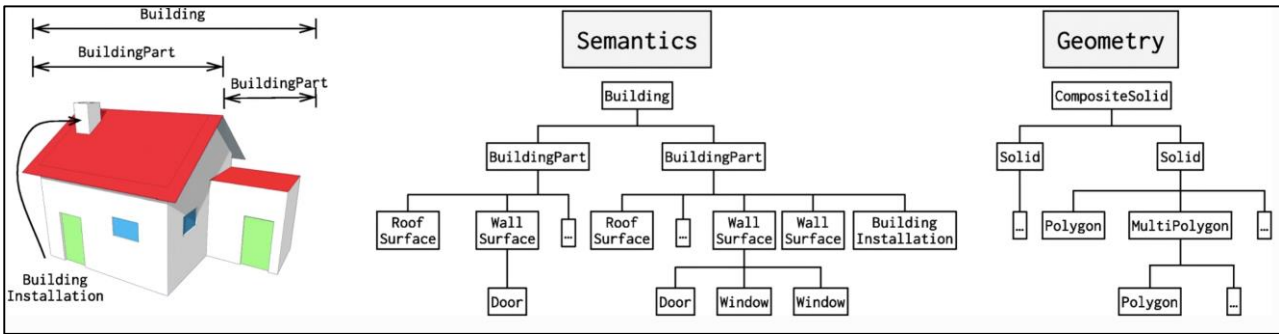
3BKM'leri ile web üzerinden etkileşime girebilmek, 3B kent modellerinin hedef kitesini, coğrafi bilgi konusunda uzman olmayan fakat çalışmalarında 3BKM'lerinden faydalanabilecek olan kitlelere genişletebilir (Prandi vd. 2015). Bu nedendir ki günümüzde pek çok kent modeli web üzerinden kullanıcılara sunulmaktadır. 3BKM'lerinin web teknolojileri ile entegrasyonu önemli bir çalışma alanı haline gelmiştir.

Bu çalışmada TKGM CityGML veri modeli ve örnek veri formatı web teknolojileri göz önünde bulundurularak incelenmiştir. Bulunan sonuçlara karşın yeni bir veri formatı önerilmiştir.

2. 3B Kent Modelleri için Veri Modelleri ve Formatları

2.1 CityGML

CityGML, 3BKM'nin depolanması ve aktarılması için tasarlanmış XML tabanlı açık bir veri modeli ve veri formatıdır. CityGML, geometrik, semantik, topolojik ve görsel özelliklerine göre kentlerdeki en alakalı topoğrafik objeler için sınıfları ve ilişkileri tanımlar (OGC CityGML 2.0, 2012). CityGML, bir kenti temsil etmek için onu daha küçük objelere böler. Kentte yaygın olarak bulunan objeler için sınıflar tanımlar ve ayrıca bu sınıflar arasındaki hiyerarşik ilişkileri tanımlar (Şekil 1).



Şekil 1. Bina objesi ve onu oluşturan daha küçük objeler (Ledoux vd. 2019)

CityGML, Special Interest Group (SIG 3D) üyeleri tarafından 2002 yılından itibaren geliştirilmeye başlandı (OGC CityGML 2.0, 2012). 2008 yılından itibaren de OGC Standardı olarak kabul edilmiştir. Kısa bir süre öncesine kadar da 3BKM'nin depolanması ve aktarılması için tek standart veri modeli ve format olması sebebiyle, bugün endüstri tarafından üretilen 3BKM'nin pek çoğu CityGML formatındadır.

CityGML, 3BKM'nin yönetilmesi noktasında yaygın bir format olsa da bu formatı destekleyen ticari ya da açık kaynak kodlu yazılımların sayısı oldukça azdır. "QGIS2Three.js" adlı bir QGIS eklentisi ile CityGML verisi QGIS'te görüntülenebilir. Bu eklenti ile CityGML okunarak Three.js adlı 3B JavaScript kütüphanesinin veri tiplerine dönüştürülerek görüntülenmektedir.

Başka pek çok çalışmada da CityGML verisi görüntüleneceği zaman öncelikle glTF, kml, collada gibi başka formatlara çevrilmiştir (Chaturvedi vd. 2015). Bunun nedeni CityGML veri yapısının doğrudan 3B veriyi görüntülemek için uygun olmamasıdır. Neden uygun olmadığı ilerleyen bölümlerde daha detaylı açıklanmıştır.

CityGML veri modelinin 3 versiyonu bulunmaktadır. Bunlar, CityGML 1.0, CityGML 2.0 ve CityGML 3.0'dır. CityGML 3.0 bu makalenin yazımı sırasında yeni standart olmuş ve geliştiriciler için kavramsal modeli yeni yayınlanmıştır. CityGML 3.0 modülü, mevcut veri modeli üzerinde pek çok büyük yenilik getirmiştir. En büyük yeniliklerden biri zamana bağlı olarak değişen verilerin temsil edilmesini sağlayan yeni "Dynamizer" modülüdür. Diğer önemli bir yenilik de yeni "Versioning" modülüdür. Bu modül ile 3BKM'de yapılan değişiklikler ve versiyon kontrolü yönetilebilecektir. CityGML 3.0 ile ilgili yenilikler için daha detaylı bilgi (Chaturvedi vd. 2017), (Chaturvedi ve Kolbe, 2017), (Kutzner ve Kolbe, 2018) ve (Kutzner vd. 2020) çalışmalarından edinilebilir.

Şüphesiz CityGML'in ön tanımlı sınıfları tüm kullanıcı senaryolarını kapsamamaktadır bu nedenle de bazı uygulamalarda, uygulamanın gerektirdiği, CityGML veri modelinde tanımlı olmayan, kullanıcı tanımlı verilerin temsil edilebilmesi için CityGML veri modelinin genişletilmesi gerekmektedir. Bu anlamda CityGML 2 mekanizma sunmaktadır. Birinci mekanizma CityGML'in "Generics" modülüdür. Bu modül sayesinde yeni 3B objeler "Generic City Objects" olarak ve kullanıcı tanımlı özneliklerde "Generic Attributes" olarak temsil edilir. İkinci mekanizma ise yeni bir Uygulama Şeması (Application Domain Extension) yani kısaca ADE geliştirmektir. ADE'lerin "Generics" modülünden farkı ise kullanıcı tanımlı obje ve özneliklerin ayrı bir XML şema tanımına sahip olmasını gerektirir. Kullanıcı tanımlı objeler, öznelikler be bunların diğer sınıflarla ilişkileri XML şema tanımlama dosyasında (XSD) tanımlanır. Böylece bu mekanizma, bu ek obje ve özneliklerim şemaya karşı validasyonuna imkan sağlamaktadır ve bu ek obje ve öznelikleri yeniden kullanılabilir yapmaktadır. ADE tanımlamasına örnek olarak gürültü ile ilgili obje ve özneliklerin tanımlandığı "noise ADE" örnek olarak verilebilir.

2.2 CityJSON

CityJSON, 3BKM'lerin depolanması ve aktarılması CityGML veri modeline dayanan, JSON ile kodlanmış yeni bir veri modeli ve formatıdır. Ayrıca bir OGC Standardıdır (OGC document 20-072r2). CityJSON veri modelinde Objeler hiyerarşisi ve geometrik veri birbirinden ayrılmıştır. Tüm objelere ait koordinatlar global bir dizide tutulmaktadır ve her bir obje için ilgili veriye, objenin indis verileri kullanılarak erişilmektedir (Şekil 2). Böylece, veri modeli minimuma indirilmiştir.

```
{
  "CityObjects": {
    "B-201391182746-80FE6BE00A9C": {
      "type": "Building",
      "geometry": [
        {
          "type": "MultiSurface",
          "boundaries": [[0,1,2],[2,3,4],[0,2,5],[4,1,7],[8,9,10],[11,12,13]
        }
      ]
    },
    "B-201391182746-80FE6BE00A9D": {
      "type": "Building",
      "geometry": [
        {
          "type": "MultiSurface",
          "boundaries": [[13,14,15],[15,16,17],[13,15,18],[17,14,20],[8,9,10]
        }
      ]
    }
  },
  "vertices": [[300762.679,5041624.941,14.274],[300749.896,5041617.688,14.274],[300755.943,5041628.123,14.274],[300769.697,5041653.169,14.274],[300762.009,5041641.543,14.274],[300760.033,5041646.854,14.274],[300779.945,5041646.854,14.274],[300779.553,5041646.725,14.274],[300779.753,5041646.725,14.274],[300780.026,5041646.431,14.274],[300780.202,5041646.694,14.274 ]],
}
```

Şekil 2. CityJSON dosyasının bir bölümü

CityJSON formatının CityGML'e göre üstünlükleri:

- JSON tabanlı olmasından ötürü XML tabanlı CityGML'e göre daha az yer kaplaması ve daha kolay, daha hızlı ayrıştırılabilmesi
- Geometrik verinin olarak sadece bir yerde global olarak tutulmasından ötürü veri tekrarını önlemesi
- JSON yapısının programlama dilleriyle, harici kütüphaneler kullanmadan ayrıştırılabilmesi
- CityJSON'ın CityGML'e göre daha web uyumlu bir format olması
- CityGML'in derin hiyerarşik yapısının CityJSON'da olmaması

3. Web Teknolojileri ve 3B Kent Modelleri ile İlişkisi

HTML5 ve WebGL gibi internet teknolojilerinin gelişmesiyle, 3B konumsal verinin herhangi bir eklenti ya da yazılım kurulumu gerektirmeden web üzerinden görselleştirilmesi mümkün hale gelmiştir. HTML5, internetin çekirdek teknolojilerinden HTML işaretleme dilinin 5. versiyonudur (Wikipedia, 2022) ve platform bağımsız web uygulamaları geliştirmek için bir açık standarttır. HTML5'in 3B WebCBS kapsamında getirdiği en büyük yenilik, web sayfalarına grafiksel içeriği eklemek ve işlemek için kullanılan yeni 'canvas' elementidir. WebGL, HTML5'in canvas elementi üzerinden koşan, telifsiz, platform bağımsız, tarayıcı bağımsız, OpenGL ES 2.0 'a dayanan bir düşük seviyeli grafik API dir (Khronos, 2014). WebGL, programcıya tarayıcı üzerinden javascript ile GPU'ya erişim imkânı vermektedir (Parisi, 2015). Böylece 3B veriyi görselleştirmek, 3B veri üzerinde operasyonlar gerçekleştirmek için GPU tarafından hızlandırılmış algoritmaların kullanımını sağlamakta ve performansı arttırmaktadır (Taraldsvik, 2011).

HTML5 ve WebGL'in birlikte kullanımıyla, bu teknolojilerden önceki uygulamaların aksine, platform ve tarayıcı bağımsız herhangi bir eklenti kurulumu gerektirmeyen pek çok web tabanlı uygulama geliştirilmiştir ve 3BKM'lerin web üzerinden görselleştirilmesi WebCBS alanında önemli bir konu başlığı haline gelmiştir. Öncelikle pek çok sanal küre uygulaması geliştirilmiştir (Chaturvedi, 2014). Cesium (Analytics Graphics Inc., 2015), WebGL Earth (Klokkan Technologies, 2011), Open Web Globe (Christen ve Nebiker, 2011) bunlardan bazılarıdır.

Sanal küreler dışında, 3BKM'lerinin web üzerinden görselleştirilmesi kapsamında da pek çok çalışma yapılmıştır. x3DOM (x3DOM, 2022) ve WebGL kullanılarak 3BKM'leri herhangi bir eklenti olmadan tarayıcıda görüntülenmiştir (Rodrigues vd. 2013). Bir başka çalışmada Java tabanlı Nasa World Wind sanal küresi 3BKM'lerini görüntüleyebilecek şekilde genişletilmiştir (Prandi vd. 2013). Mao ve Ben (2011) tarafından yapılan çalışmada, 3BKM, bina, cadde gibi kent objelerini temsil eden Java sınıflarına dönüştürülmüş ve x3DOM kullanılarak görüntülenmiştir. Prieto vd. (2012), bu çalışmayı genişleterek Java sınıfları yerine W3DS gibi 3B web servisleri kullanarak 3BKM tarayıcı üzerinden görüntülenmiştir. Bu kapsamdaki benzer çalışmalar için ilgili diğer kaynaklara bakılabilir (Schilling vd. 2012; Gesquiere ve Manin, 2012; Chaturvedi, 2014; Prandi vd. 2013; Prandi vd. 2015; Chaturvedi vd. 2015; Pispidikis ve Dimopolou, 2016).

4. 3B Kadastro Çalışmaları ve TKGM CityGML Veri Modeli

4.1 3B Kadastro Çalışmaları

3B kadastro çalışmaları, ülke genelinde TKGM tarafından yürütülen "3B Kent Modelleri ve Kadastro" projesi ile yürütülmektedir. Bu çalışma kapsamında, kentlerin 3BKM'leri üretilmekte ve 3B kadastro altlıkları hazırlanmaktadır. Bu iş için ihale yoluyla özel sektör tarafından yapılmaktadır. 3B kent modelleri fotogrametrik olarak, yapılan uçuşlar ile üretilmektedir. Bağımsız bölümler, mimari projeler kullanılarak manuel olarak yüklenici firmalar tarafından çizilmekte ve oluşturulan 3B modeller 3BKM'ne yine yüklenici tarafından entegre edilmektedir. Yüklenici firmalar tarafından hazırlanan veriler TKGM'ne TKGM CityGML formatında teslim edilmektedir.

4.2. TKGM CityGML Veri Modeli

TKGM, 3B Kadastro çalışmasında "Generics" modülünü kullanarak mevcut CityGML veri modelini genişletmiş, yeni veri modelinin adına da "TKGM CityGML" demiştir. Bu veri modeline göre, mevcut veri modeline eklenen objeler ve sınıf isimleri Tablo 1 de gösterilmektedir. Burada en çok dikkat çeken bağımsız bölüm objesidir. CityGML veri modelinde, bina sınıfında bağımsız bölüm olmadığı için bağımsız bölümlerin geometrileri "GenericCityObjects" olarak mevcut CityGML veri modeline eklenmiştir. Bağımsız bölümlere ait öznitelik verileri de "GenericAttributes" olarak tutulmaktadır. Bu verilerin neler olması gerektiğine ilişkin veriler Tablo 2 de yer almaktadır.

Tablo 1. TKGM veri modelindeki sınıflar ve isimleri (3D Kadastro, 2022)

Tür	Değer
Mimari Bina	MimariBina
Fotogrametrik Bina	FotogrametrikBina
Fotogrametrik Bina Part	FotogrametrikBinaPart
Bina Çatı Eklenti	BinaCatiEklenti
Kat	Kat
Bağımsız Bölüm	BagimsizBolum
Bağımsız Bölüm Kısım	BagimsizBolumKisim
Bağımsız Bölüm Kısım İç Yapı	BagimsizBolumKisimIcYapi
Ortak Alan İç Yapı	OrtakAlanIcYapi
Balkon	Balkon
Teras (Mimari Bina)	Teras
Ortak Alan	OrtakAlan
Mimari Bina Grup	MimariBinaGrup

Tablo 2. Bağımsız bölüm sınıfına ait öznitelik verileri

Özellik Adı	Generic Attribute Tipi	Zorunluluk
geometrySuitability (Geometrik Uygunluk)	gen:intAttribute	Evet
integrationState (Entegrasyon Durumu)	gen:intAttribute	Evet
independentSectionNumber (Bağımsız Bölüm Numarası)	gen:stringAttribute	Evet
takbisPropertyIdentityNumber (Taşınmaz Id)	gen:intAttribute	Evet
additionalNote (Ek Açıklama)	gen:stringAttribute	Hayır
independentSectionCardinalDirection (Bağımsız Bölüm Cephe Bilgisi)	gen:intAttribute	Evet
independentSectionUsage (Bağımsız bölüm kullanım amacı)	gen:intAttribute	Evet
partCount (Bağımsız Bölüm Kısım Adedi)	gen:intAttribute	Evet
independentSectionPlanNetArea (Bağımsız Bölüm Mimari Projede Yazan Net Alan)	gen:doubleAttribute	Evet
independentSectionPlanGrossArea (Bağımsız Bölüm Mimari Projede Yazan Brüt Alan)	gen:doubleAttribute	Evet
independentSectionCalculatedArea (Bağımsız Bölüm Çizim Alan)	gen:doubleAttribute	Evet
blockName (Blok Numarası)	gen:stringAttribute	Evet
entrance (Giriş)	gen:stringAttribute	Evet

4.3. TKGM CityGML Veri Modelinin CityGML Kaynaklı Eksiklikleri

CityGML, her ne kadar popüler olarak kullanılsa da pek çok eksiklik barındırmaktadır. Verilerin hangi şekilde depolanabileceğini, düzenlenebileceğini ve manipüle edilebileceğini belirleyen farklı veri modelleri vardır. En yaygın veri modeli yaklaşımlarından biri hiyerarşik veri modelidir. Hiyerarşik modelde veriler, her kaydın tek bir ebeveyni veya kökü olduğu ağaç benzeri bir yapı halinde düzenlenir. Her kayıt, geometrisi ve öznitelikleri ile saklanır. Ortak veriler, her kayıt için tekrar tekrar depolanır. Hiyerarşik veri modellerinde, veri aranması, gerekli veri bulunana kadar tüm modelin yukarıdan aşağıya araştırılmasını gerektirdiğinden, hiyerarşinin en altındaki verilere erişim yavaşken, hiyerarşinin en üstündeki verilere erişim hızlıdır. CityGML de hiyerarşik veri modeli yaklaşımını benimsemektedir ve hiyerarşik veri modelinin dezavantajlarına sahiptir. Öncelikle CityGML'in dolayısıyla TKGM CityGML'in çok karmaşık bir hiyerarşiye sahiptir. CityGML de bir obje pek çok alt objeden, her bir alt obje de başka objelerden oluşmaktadır. Bu durum CityGML'in ayrıştırılması için programlama dillerinde pek çok iç içe döngü yazılmasına sebep olmakta, bu durum, ayrıştırma performansını olumsuz olarak etkilemekte ve ayrıştırma işini zorlaştırmaktadır.

Diğer bir önemli sorun veri tekrarıdır. Hiyerarşik veri modellerinde olduğu gibi CityGML ve TKGM CityGML de her kayıt geometrisi ve öznitelikleriyle ayrı ayrı tutulduğu için, ortak geometriye sahip objelerin kayıtlarında geometrilerin ortak kısımları gereksiz yere tekrar etmiş olur. Şekil 3'deki örnek CityGML verisinde, aynı koordinatlara sahip farklı kayıtlar için, verilerin tekrar ettiği görülmektedir. Biljecki vd. (2015a) yaptıkları çalışmayla veri tekrarı problemini özellikle multi-LOD bağlamında çözmeye çalışmışlardır. Geliştirdikleri metodolojiyle birbiri ile ilişkili objeler kent modeli içinde belirlenerek aralarında tekrar eden veriler için xlink referansları oluşturulmaktadır. Böylece veri tekrarı minimuma indirgenmiştir. Ancak bir xlink referansı oluşturulabilmesi için öncelikle bir objenin oluşturulması gerekmektedir ve CityGML de bir objenin yüzeylerini ilk kez oluştururken bile poligon geometrilerinin koordinat dizileri içinde veri tekrarı olmaktadır (Şekil 2). Bu nedenle bu yöntem veri tekrarı problemini tamamen çözememiştir.

```
<gml:surfaceMember>
  <gml:Polygon gml:id="Wall-85c62d0c-cceb-40c2-8f59-ab31ee16aa30">
    <gml:exterior>
      <gml:LinearRing>
        <gml:posList>618499.087052122 4100087.70435038 942.359982833862 618499.087052122
      </gml:LinearRing>
    </gml:exterior>
  </gml:Polygon>
</gml:surfaceMember>
<gml:surfaceMember>
  <gml:Polygon gml:id="Wall-8cbd98ba-1de4-4b7b-9fc3-e4ab646c161b">
    <gml:exterior>
      <gml:LinearRing>
        <gml:posList>618499.039479032 4100087.34239665 942.359982833862 618499.039479032
      </gml:LinearRing>
    </gml:exterior>
  </gml:Polygon>
</gml:surfaceMember>
<gml:surfaceMember>
  <gml:Polygon gml:id="Wall-3fa074e3-d128-4d93-b0c7-350e0c754d39">
    <gml:exterior>
      <gml:LinearRing>
        <gml:posList>618496.205796972 4100085.41456071 942.359982833862 618496.205796972
      </gml:LinearRing>
    </gml:exterior>
  </gml:Polygon>
</gml:surfaceMember>
```

Şekil 3. CityGML veri dosyasının bir kısmı

TKGM CityGML in bir diğer önemli eksikliği de JSON yerine XML tabanlı olmasıdır (Şekil 4). Günümüzde JSON web teknolojileri tarafından kullanılan en yaygın formattır. Bunun nedeni XML'e göre çok daha kompakt olması, az yer kaplaması ve içeriğinin çok daha kolay ayrıştırılabilmesidir. Pek çok dilde yerleşik olarak JSON'daki "anahtar değer" çiftlerine karşılık gelen veri tipleri mevcuttur. Örneğin Java ve Go dillerindeki "Map" veri tipi, Python'daki "Dictionary" veri tipi buna örnek olarak gösterilebilir. Böylece, programlama dilleriyle herhangi bir harici kütüphane kullanılmadan JSON dokümanları bu veri tiplerine eşlenebilmektedir. Oysa XML de durum daha karmaşıktır. Pek çok dilde XML dokümanlarını ayrıştırabilmek için harici kütüphaneler kullanılmak zorundadır ve bir XML dokümanını ayrıştırmak için kompleks "marshalling" ve "unmarshalling" işlemleri gerekmektedir.

CityGML dokümanları da XML tabanlı olduğundan yukarıda bahsedilen eksiklikleri aynen taşımaktadırlar. TKGM CityGML veri modeli de XML tabanlı olduğundan ve CityGML e dayandığından aynı eksiklikler TKGM CityGML veri modeli için de geçerlidir.

Üçüncü önemli eksiklik de CityGML de ve dolayısıyla TKGM CityGML de objelerin geometrik, semantik ve özniteliklerinin bir arada iç içe barındırmasıdır. Örneğin bir binanın geometrisi de öznitelikleri de "Building" etiketi altında tutulmaktadır. Söz konusu bir görselleştirme uygulamasında sadece geometrik veriye ihtiyaç olmasına karşın bina etiketi altındaki tüm etiketler araştırılarak sadece geometri içerenler ayrıştırılmaktadır. Bu ekstra ayrıştırma işlemi de koşum zamanı performansını olumsuz etkilemektedir.

4.4 TKGM CityGML Veri Modelinin Tasarım Açısından Eksiklikleri

Bağımsız Bölümler "generics" modülü kullanılarak CityGML veri modeline eklendiğinden, her bir bağımsız bölüm bir jenerik kent objesidir "Generic City Object" (Şekil 3). Jenerik objeler CityGML hiyerarşisinde kent objesi seviyesindedir yani bir bina objesi ile eş değerdir. Oysa bağımsız bölümler binaların içinde yer alan objelerdir ve hiyerarşide binaların altında yer almalıdırlar. Dolayısıyla bağımsız bölümler binalar ile ilişkilendirilemediğinden bir binaya ait bağımsız bölümler için sorgu yapılamamaktadır. Herhangi bir binaya ait bağımsız bölüm için tüm binaların bağımsız bölümleri araştırılmaktadır. Oysa CityGML veri modeli ADE kullanılarak genişletilmeli ve bağımsız bölümler bina sınıfının alt objeleri olarak tanımlanmalıydı, böylece binalar ile bağımsız bölümler ilişkilendirilebilecek ve sorgular çok daha etkin bir şekilde yapılabilecekti.

```
<core:cityObjectMember>
  <bldg:Building gml:id="MB_ab57b89d-349f-4b0a-a498-ed322efdbf8b"> ...
</bldg:Building>
</core:cityObjectMember>
<core:cityObjectMember>
  <gen:GenericCityObject gml:id="BB_f2ff4ed9-314c-45f7-87d1-ac51e4e7ebfe">
    <gml:name>Bağımsız Bölüm 21</gml:name>
    <gen:stringAttribute name="independentSectionNumber">
      <gen:value>21</gen:value>
    </gen:stringAttribute>
    <gen:stringAttribute name="takbisPropertyIdentityNumber">
      <gen:value>8FBCE29564CA9</gen:value>
    </gen:stringAttribute>
    <gen:stringAttribute name="additionalNote">
      <gen:value>-</gen:value>
    </gen:stringAttribute>
    <gen:intAttribute name="partCount">
      <gen:value>1</gen:value>
    </gen:intAttribute>
    <gen:intAttribute name="geometrySuitability">
      <gen:value>1001</gen:value>
    </gen:intAttribute>
    <gen:intAttribute name="integrationState">
      <gen:value>1001</gen:value>
    </gen:intAttribute>
  </gen:GenericCityObject>
</core:cityObjectMember>
```

Şekil 4. Örnek TKGM CityGML dosyasının bir bölümü

Bir diğer tasarım eksikliği de geometrinin depolanış şeklidir. Geometrik veri, sıralı koordinat dizileri şeklinde tutulmaktadır (Şekil 5). Bu durum veri tekrarına sebep olmakla birlikte başka sorunları da beraberinde getirmektedir.

```
</bldg:lod2Geometry>
<bldg:boundedBy>
  <bldg:FloorSurface>
    <bldg:lod2MultiSurface>
      <gml:MultiSurface srsName="EPSG:5256">
        <gml:surfaceMember>
          <gml:Polygon gml:id="Floor-1c79dd5e-7887-4c15-85b7-bbd439d6a34f">
            <gml:exterior>
              <gml:LinearRing>
                <gml:posList>618498.195370451 4100098.90219346 930.559989080429 618498.1
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
        </gml:surfaceMember>
        <gml:surfaceMember>
          <gml:Polygon gml:id="Floor-49e761a1-2e8a-4bfa-b29b-2265696bfa83">
            <gml:exterior>
              <gml:LinearRing>
                <gml:posList>618496.662863508 4100103.53566018 930.359986886978 618496.6
              </gml:LinearRing>
            </gml:exterior>
          </gml:Polygon>
        </gml:surfaceMember>
        <gml:surfaceMember>
          <gml:Polygon gml:id="Floor-a35125f7-6332-42a8-95ba-05edc3b106e8">
            <gml:exterior>
              <gml:LinearRing>
```

Şekil 5. TKGM CityGML dosyasında geometrik verinin depolanması

Web uygulamalarında verinin görselleştirilmesi WebGL grafik API'si kullanılarak yapılmaktadır. WebGL, görüntülenecek veriyi "vertex" dizisi ve "index" dizisi şeklinde istemektedir. Bu sebepten ötürü, her bir objenin koordinat dizisi ayrıştırılarak indekslenmesi gerekmektedir. 2B veriler için bu durum söz konusu değildir. Örneğin GeoJSON dosyalarında da geometri sıralı koordinat dizileri olarak tutulmaktadır. Ancak, 2B verinin görselleştirilmesi için kullanılan Canvas API zaten doğrudan sıralı koordinat dizileriyle çalıştığından böyle bir işleme gerek yoktur. Oysa TKGM CityGML örneğinde verinin görselleştirilebilmesi için ek bir indeksleme işlemine ihtiyaç vardır.

5. Önerilen Veri Modeli ve Veri Formatı

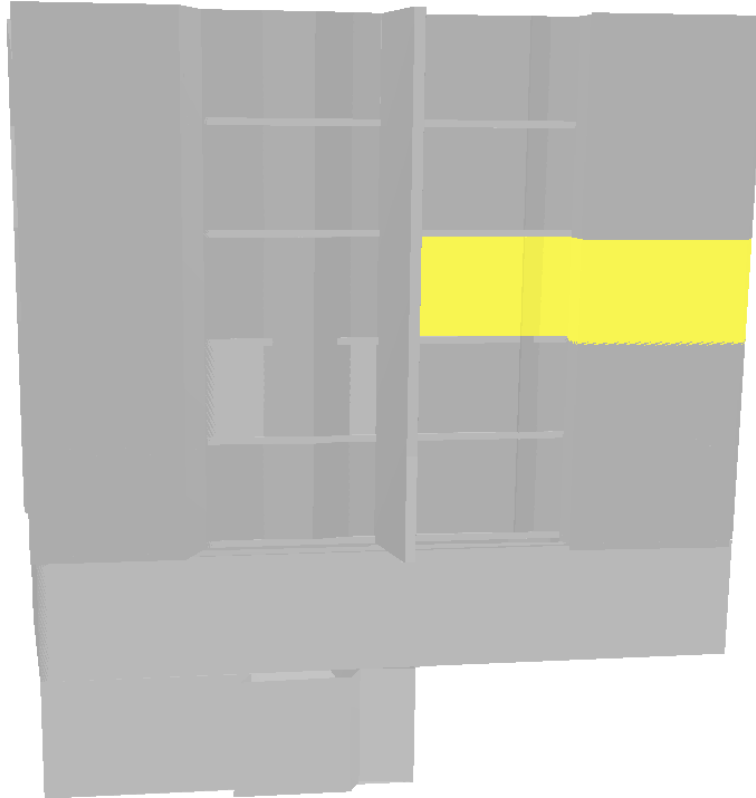
Bu çalışmada, TKGM CityGML veri modeli ve formatının eksikliklerini giderecek yeni bir veri modeli tasarlanmıştır. Öncelikle, XML tabanlı eksiklikleri gidermek için JSON tabanlı bir format önerilmiştir. Önerilen format CityJSON formatına dayanmaktadır (Ledoux vd. 2019). CityJSON, CityGML veri modeli için geliştirilmiş bir JSON tabanlı veri değişim formatıdır. Bu formatın seçilmesindeki en büyük etkenlerden biri de geometrik verinin her bir kayıt için sıralı koordinat dizileri şeklinde değil de global bir vertex dizisi ve her bir kayıt için index dizisi şeklinde tanımlanmış olmasıdır. Böylece görselleştirme aşamasında veri grafik API'lara gönderilirken ek bir indeksleme işlemi gerekmemektedir.

CityJSON doğrudan CityGML veri modelini kullandığından, CityGML veri modelinde olmayan bağımsız bölüm objeleri, CityJSON veri modelinde de bulunmamaktadır. Bağımsız bölümleri depolamak için CityJSON genişletilmiş ve bina objelerinin için bağımsız bölüm objeleri eklenmiştir (Şekil 6). Böylece hangi bağımsız bölümün hangi binaya ait olduğu bilgisi veri ayrıştırılırken kolayca elde edilebilmektedir. Şekil 6 da "B-206391182646-3B58D3D11C14" id'li bir binanın koordinatları "boundaries" anahtarının değerleri olarak verilmekte, bu binaya ait bağımsız bölümler de "condominiums" anahtarının değeri olarak geometrisi ile birlikte verilmektedir. Geometrilere, global vertex listesini gösteren indeks değerlerinden oluştuğundan, veri tekrarı olmamakta ve ek indeksleme olmadan veri grafik API'ler ile render edilebilmektedir.


```
"CityObjects": {
  "B-201391182746-80FE6BE00A9C": {
    "type": "Building",
    "geometry": [
      {
        "type": "MultiSurface",
        "boundaries": [[[0,1,2]], [[2,3,4]], [[0,2,5]], [[4,1,7]], [[8,9,10]], [[11,12,13]]
      }
    ]
    "condominiums": [
      {
        "id": "C-201391182746-80FE6BE00A9C",
        "type": "MultiSurface",
        "boundaries": [[[0,1,2]], [[2,3,4]], [[0,2,5]], [[4,1,7]]]
      },
      {
        "id": "C-201391182746-80FE6BE00A9C",
        "type": "MultiSurface",
        "boundaries": [[[8,9,10]], [[11,12,13]], [[9,14,15]]]
      }
    ]
  }
}
```

Şekil 6. Örnek CityJSON dosyası bina detayı ve bağımsız bölüm detayları

Bağımsız bölümleri de içeren CityJSON dosyası görüntülenmiş (Şekil 7) ve TKGM CityGML dosyası ile karşılaştırılmıştır. Yapılan testler sonucu örnek TKGM CityGML dosyası 4.80mb iken üretilen CityJSON dosyasının 0.91mb olduğu görülmüştür. Buradan da önerilen veri modelinin veri kaybı olmaksızın yaklaşık 5 kat daha az yer kapladığı görülmüştür.



Şekil 7. Bağımsız bölümlerin CityJSON olarak görüntülenmesi

5. Sonuç

Bu çalışmada, ülkemizde yürütülen 3B kadastro çalışmalarına altlık teşkil eden TKGM CityGML veri modeli ve örnek veri dosyası bir ilk olarak incelenmiştir. Yapılan çalışmalarda TKGM CityGML veri modelinin önemli eksiklikler içerdiği görülmüştür. TKGM CityGML veri modelinde veriyi ayrıştırmanın zor olduğu özellikle hiyerarşik olarak aynı seviyede olmaması gereken bina ve bağımsız bölüm gibi objelerin aynı seviyede depolanmasının pek çok sorun teşkil ettiği görülmüştür. TKGM CityGML verilerinin mevcut haliyle veri tekrarı içerdiği, doğrudan görselleştirilemediği, görselleştirilmesi için ek işlemler gerektirdiği görülmüştür. Tüm bu eksikliklere karşın CityJSON veri formatına dayanan yeni bir veri formatı önerilmiştir. Ayrıca TKGM CityGML dosyasını ayrıştırmak için Go dilinde AKK bir ayrıştırıcı (parser) geliştirilmiştir. Bu ayrıştırıcı go dilinde yazılmış ilk CityGML ayrıştırıcısıdır ve açık kaynak versiyonuna github/ZiyaUsta adresinden ulaşılabilir.

Kaynaklar

- Analytical Graphics Inc. (2015). *AGI An Ansys Company*, Retrieved from <https://www.agi.com/home>
- Biljecki, F., Ledoux, H., & Stoter, J. (2015). Improving the consistency of multi-LOD CityGML datasets by removing redundancy. *3D Geoinformation Science*, (pp. 1-17). Springer International Publishing, Cham.
- Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., & Çöltekin, A. (2015). Applications of 3D city models: State of the art review. *ISPRS International Journal of Geo-Information*, 4(4), 2842-2889.
- Chaturvedi, K. (2014). *Web based 3D analysis and visualization using HTML5 and WebGL* (Master's thesis), University of Twente, Faculty of Geo-Information Science and Earth Observation, Netherlands.
- Chaturvedi, K., & Kolbe, T. H. (2017). *Future City Pilot 1 Engineering Report* (Report No. 16-098). Lehrstuhl für Geoinformatik.
- Chaturvedi, K., Willenborg, B., Sindram, M., & Kolbe, T. H. (2017). Solar potential analysis and integration of the time-dependent simulation results for semantic 3D city models using dynamizers. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-4/W5(2017), 25-32.
- Chaturvedi, K., Yao, Z., & Kolbe, T. H. (2015). Web-based Exploration of and interaction with large and deeply structured semantic 3D city models using HTML5 and WebGL. In *Wissenschaftlich-Technische Jahrestagung der DGPF und Workshop on Laser Scanning Applications*. 2015. (pp. 296-306). DGPF Tagungsband 24/2015.
- Christen, M., & Nebiker, S. (2011). OpenWebGlobe SDK-An open source high-performance virtual globe SDK for open maps. In *1st European State of the Map Conference*, Vienna.
- Gesquière, G., & Manin, A. (2012). 3D visualization of urban data based on CityGML with WebGL. *International Journal of 3-D Information Modelling (IJ3DIM)*, 1(3), 1-15.
- Khronos, (2014), *Khronos Group, WebGL Specification, Version 1.0.3.*, Retrieved from <https://www.khronos.org/registry/webgl/specs/latest/1.0/>
- Klokan Technologies, (2011, Aralık 20). *WebGL Earth- open source 3D digital globe written in JavaScript*, Retrieved from <http://www.webglearth.org/>
- Kutzner, T., Chaturvedi, K., & Kolbe, T. H. (2020). CityGML 3.0: New functions open up new applications. *PFG-Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, 88(1), 43-61.
- Kutzner, T., & Kolbe, T. H. (2018). CityGML 3.0: sneak preview. In *PFGK18-Photogrammetrie-Fernerkundung-Geoinformatik-Kartographie, 37. Jahrestagung in München 2018*, 835-839.
- Ledoux, H., Arroyo Ogori, K., Kumar, K., Dukai, B., Labetski, A., & Vitalis, S. (2019). CityJSON: A compact and easy-to-use encoding of the CityGML data model. *Open Geospatial Data, Software and Standards*, 4(1), 1-12.
- Mao, B., & Ban, Y. (2011). Online Visualisation of a 3D City Model Using CityGML and x3DOM. *Cartographica*, 46(2), 109-114.
- OGC CityGML 2.0, (2012), *OGC city geography markup language (CityGML) encoding standard* (Report No. 12-019). Open Geospatial Consortium Version 2.0.0.
- Pispidikis, I. & Dimopoulou, E, (2016, October). Development of a 3D WebGIS system for retrieving and visualizing CityGML data based on their geometric and semantic characteristics by using free and open source technology, *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, In *11th 3D Geoinfo Conference*, IV-2/W1(2016), (pp. 47-53).
- Prandi, F., De Amicis, R., Piffer, S., Soave, M., Cadzow, S., Boix, E. G., & D'Hondt, E. (2013, May). Using CityGML to deploy smart-city services for urban ecosystems. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XL-4/W1, 29th Urban Data Management Symposium*, 2013. (pp. 87-92).
- Prandi, F., Devigili, F., Soave, M., Di Staso, U. & De Amicis, R. (2015). 3D web visualization of huge CityGML models. In *ISPRS Geospatial Week 2015. Proceedings*, XL-3/W3(2015), 601-605.

- Prieto, I., Izkara, J. L., & Delgado del Hoyo, F. J. (2012, June). Efficient visualization of the geometric information of CityGML: application for the documentation of built heritage. In *International Conference on Computational Science and Its Applications* (pp. 529-544).
- Rodrigues, J. I., Figueiredo, M. J., & Costa, C. P. (2013, July). Web3D GIS for city models with CityGML and X3D. In *2013 17th International Conference on Information Visualisation* (pp. 384-388).
- Schilling, A., Hagedorn, B., & Coors, V. (2012). OGC 3D Portrayal Interoperability Experiment Final Report. *Open Geospatial Consortium Engineering Report*.
- Taraldsvik, M. (2011). Exploring the Future: is HTML5 the solution for GIS Applications on World Wide Web? Technical Report, NTNU.
- Parisi T. (2015). Programming 3D Applications with HTML5 and WebGL, O'Really Media.
- Wikipedia, (2022, Ocak 8), *HTML 5*, Retrieved from <https://tr.wikipedia.org/wiki/HTML5>
- x3DOM, (2022, Ocak 4), *x3DOM instant 3D the HTML way*. Retrieved from <https://www.x3dom.org/>
- 3D Kadastro (2022, Ocak 5), *GML ID, Name, Class Atamaları*. Retrieved from http://cbs.tkgm.gov.tr/3d/html/gml_id_name_class_atamaları.html