

**T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**BİNA NESNELERİNİN CITYGML STANDARDINA GÖRE LoD2 VE LoD3
AYRINTI DÜZEYLERİNDE ÜÇ BOYUTLU MODELLENMESİ**

ŞÜHEDA ÖZDOĞAN

**YÜKSEK LİSANS TEZİ
HARİTA MÜHENDİSLİĞİ ANABİLİM DALI
UZAKTAN ALGILAMA VE CBS PROGRAMI**

**DANIŞMAN
DOÇ. DR. ALİ MELİH BAŞARANER**

İSTANBUL, 2014

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**BİNA NESNELERİNİN CITYGML STANDARDINA GÖRE LoD2 VE LoD3
AYRINTI DÜZEYLERİNDE ÜÇ BOYUTLU MODELLENMESİ**

Şüheda ÖZDOĞAN tarafından hazırlanan tez çalışması 12.03.2014 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Harita Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Doç. Dr. Ali Melih BAŞARANER
Yıldız Teknik Üniversitesi

Jüri Üyeleri

Doç. Dr. Ali Melih BAŞARANER
Yıldız Teknik Üniversitesi

Prof. Dr. N. Necla ULUĞTEKİN
İstanbul Teknik Üniversitesi

Yrd. Doç. Dr. Fatih GÜLGEN
Yıldız Teknik Üniversitesi

ÖNSÖZ

“Bina Nesnelerinin CityGML Standardına Göre LoD2 ve LoD3 Ayrıntı Düzeylerinde Üç Boyutlu Modellenmesi” adlı bu çalışma, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Harita Mühendisliği Anabilim Dalında yüksek lisans tezi olarak hazırlanmıştır.

Öncelikle tez çalışması süresince bilgi ve tecrübelerinden faydalandığım, kendisinden her zaman destek ve anlayış gördüğüm, tezimin her aşamasında yardımını esirgemeyen ve önerileri ile yol gösteren tez danışmanım ve değerli hocam Doç. Dr. Ali Melih BAŞARANER’e teşekkürlerimi sunarım.

Tez çalışmamda yazılım desteği nedeniyle UVM Systems GmbH’a ve uygulama aşamasındaki teknik destek için proje yöneticisi Alpaslan TÜTÜNEKEN’e (MSc. Cartography and Geoinformation, University of Vienna) teşekkürlerimi sunarım.

Çalışmam sırasında manevi desteğini ve yardımını esirgemeyen sayın hocam Doç. Dr. Füsün BALIK ŞANLI’ya teşekkürlerimi sunarım.

Tez süresince bana göstermiş oldukları anlayış ve destek için başta Aynur ZONTUL ve Nihal ÖNER olmak üzere tüm arkadaşlarıma teşekkür ederim. Her türlü sıkıntıda bana olan desteğini hiçbir zaman esirgemeyen sevgili annem Ayşe ÖZDOĞAN ve sevgili babam Alaaddin ÖZDOĞAN’a ve kardeşlerim İbrahim ÖZDOĞAN ve Nida ÖZDOĞAN’a en içten teşekkürlerimi sunarım.

Mart, 2014

Şüheda ÖZDOĞAN

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ.....	viii
KISALTMA LİSTESİ.....	ix
ŞEKİL LİSTESİ.....	xi
ÇİZELGE LİSTESİ	xiii
ÖZET	xiv
ABSTRACT.....	xvi
BÖLÜM 1	
GİRİŞ	1
1.1 Literatür Özeti	1
1.2 Tezin Amacı	4
1.3 Hipotez	4
BÖLÜM 2	
3B KENT MODELLEME.....	5
2.1 3B Mekansal Modellerin Temsili İçin Veri Yapıları.....	5
2.1.1 Sınır Temsili (B-Rep) Modeli	5
2.1.2 Yapısal Katı Geometri (CSG)	7
2.1.3 Parametrik Model	8
2.2 Yaygın 3B Mekansal Veri Toplama Yöntemleri	9
2.2.1 Lazer Tarama	9
2.2.2 Fotogrametrik Ölçme Yöntemi.....	10
2.2.3 GNSS Verileri	11
2.3 3B Kent Modellerinin Kullanım Alanları.....	11
2.3.1 3B Kent Modeli Oluşturma ve Yönetimi	12
2.3.2 3B Kent Modeli Görselleştirme	13
2.4 3B Modellemede Kullanılan Standartlar.....	13
2.5 3B Mekansal Modellemeye İlişkin Genel Problemler	15

BÖLÜM 3

CityGML STANDARDI.....	17
3.1 CityGML' e Genel Bakış	17
3.1.1 Tarihsel gelişimi	18
3.1.2 CityGML 2.0'daki Ek Özellikler	19
3.2 CityGML'in Genel Özellikleri	20
3.2.1 Ayrıntı Düzeyleri (LoD).....	20
3.2.2 Modülleştirme	21
3.2.3 Kapatma Yüzeyi.....	22
3.2.4 Arazi Kesişim Eğrisi (TIC).....	22
3.2.5 Görünüşler	23
3.2.6 Uygulama Alanı Eklentileri (ADE).....	23

BÖLÜM 4

CityGML'DE MODELLEME ANLAYIŞI VE YAPISI	24
4.1 Tematik Model	24
4.2 Semantik Model	26
4.3 Geometrik ve Topolojik Model	27
4.4 Görünüş Modeli	29

BÖLÜM 5

CityGML'DE BİNA MODELİ	30
5.1 Binaların Farklı Ayrıntı Düzeylerinde (LoD) Modellenmesi.....	35
5.1.1 Ayrıntı Düzeyi 0 (LoD0)	38
5.1.2 Ayrıntı Düzeyi 1 (LoD1)	39
5.1.3 Ayrıntı Düzeyi 2 (LoD2)	40
5.1.4 Ayrıntı Düzeyi 3 (LoD3)	42
5.1.5 Ayrıntı Düzeyi 4 (LoD4).....	44
5.2 INSPIRE ve CityGML Bina Modelleri Arasındaki Farklar	45

BÖLÜM 6

UYGULAMA	47
6.1 Veriler.....	47
6.2 Yazılım	47
6.2.1 CityGRID	49
6.2.1.1 CityGRID'de Bina Semantiği.....	50
6.2.1.2 CityGRID'de Farklı Ayrıntı Düzeylerini (LoD) Modelleme	52
6.3 Gerçekleştirme.....	53
6.3.1 Format Dönüşümü	53
6.3.2 Koordinat Dönüşümü ve Boyutlandırma	56
6.3.3 Geometrik Modelleme ve İşleme	57
6.3.4 LoD 2 ve LoD 3 Ayrıntı Düzeylerinde Semantik Modelleme	58
6.3.5 3B Modeller için Doku Giydirme	59
6.3.6 CityGML Formatına Dönüşüm	60
6.4 Bulgular	60

6.5 Sorunlar.....	62
BÖLÜM 7	
SONUÇ VE ÖNERİLER	65
KAYNAKLAR.....	67
EK-A	
UML NOTASYONU	71
EK-B	
CityGML BİNA MODELİNİN UML DİYAGRAMI.....	72
ÖZGEÇMİŞ	74

SİMGE LİSTESİ

σ 3B nokta koordinatlarının standart sapması

KISALTMA LİSTESİ

2B	2 Boyutlu
2.5D	2.5 Dimensional
2.5B	2.5 Boyutlu
3B	3 Boyutlu
3D	3 Dimensional
3D PDF	3 Dimensional Portable Document Format
ADE	Application Domain Extension
BIM	Building Information Modeling
B-Rep	Boundary Representation
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CBS	Coğrafi Bilgi Sistemi
CityGML	City Geography Markup Language
COLLADA	Collaborative Design Activity
CSG	Constructive Solid Geometry
DXF	Data Exchange Format
GDI NRW	Geodata Infrastructure North Rhine-Westphalia
GIS	Geographic Information System
GML	Geography Markup Language
GNSS	Global Navigation Satellite System
GPS	Global Position System
HDS	High Definition Survey
IFC	Industry Foundation Classes
INS	Inertial Navigation System
INSPIRE	Infrastructure for Spatial Information in the European Community
ISO	International Organization for Standardization
KML	Keyhole Markup Language
LiDAR	Light Detection and Ranging
LoD	Level of Detail
MRDB	Multiple Representation Database
OASIS	Organization for the Advancement of Structured Information Standards
OGC	Open Geospatial Consortium
OWS	OpenGIS Web Services
SAM	Sayısal Arazi Modeli

SHP	Shape File
SIG 3D	Special Interest Group 3D
TIC	Terrain Intersection Curve
TIN	Triangulated Irregular Network
TUCBS	Türkiye Ulusal Coğrafi Bilgi Sistemi
TWG BU	Thematic Working Group on Building
U3D	Universal 3 Dimensional
UML	Unified Modeling Language
UVM	Urban Visualisation and Management
VGE	Virtual Geographic Environment
VR	Virtual Reality
VRML	Virtual Reality Modeling Language
X3D	Extensible 3D
XAL	Extensible Address Language
XML	Extensible Markup Language
XNAL	Extensible Name and Address Language
XSD	XML Schema Definition

ŞEKİL LİSTESİ

	Sayfa
Şekil 2. 1 B-Rep'e ilişkin düzlemsel çokgen temsili	6
Şekil 2. 2 3B bir nesnenin 2B yüzeyler ile B-Rep temsili.....	6
Şekil 2. 3 CSG ile modellenmiş 3B nesne.....	7
Şekil 2. 4 Parametrik bina modeli.....	8
Şekil 2. 5 Lazer tarama - nokta bulutu.....	10
Şekil 2. 6 Berlin 3B kent modeli.....	12
Şekil 3. 1 Bir binanın dört farklı ayrıntı düzeyinde temsili	21
Şekil 3. 2 Yaya alt geçidi ve tünel için kapatma yüzeyi temsili	22
Şekil 3. 3 Bir binanın ve tünelin arazi ile arakesitleri.....	23
Şekil 4. 1 Bitişğinde garaj olan bir bina.....	28
Şekil 5. 1 CityGML'de bina modeli.....	31
Şekil 5. 2 Bir ve iki bina parçası içeren bina örneği	33
Şekil 5. 3 Bina dış iskeletinin sınır yüzeyi.....	35
Şekil 5. 4 Sınır yüzeylerinin özellikle açıklıklar için sınıflandırılması.....	35
Şekil 5. 5 CityGML tarafından sağlanan beş ayrıntı düzeyi (LoD).....	36
Şekil 5. 6 LoD0'da bina temsili: geometrik gösterim ve UML örnek diyagramı	38
Şekil 5. 7 LoD0'da CityGML bina modeli (taban alanı ve çatı kenarı) ve arazi modeli .	38
Şekil 5. 8 LoD1'de bina temsili: geometrik temsil ve UML örnek diyagramı	39
Şekil 5. 9 LoD1'de CityGML bina modeli: 3B grafik ve model hiyerarşisi.....	40
Şekil 5. 10 LoD2'de bina temsili: geometrik temsil ve UML örnek diyagramı	41
Şekil 5. 11 Bina modelinin sınırındaki iki ek tematik yüzey türü	41
Şekil 5. 12 LoD2'de CityGML bina modeli	42
Şekil 5. 13 LoD3'de bina temsili: geometrik temsil, UML örnek diyagramı	43
Şekil 5. 14 LoD3'de CityGML bina modeli 3B geometri ve model hiyerarşisi	43
Şekil 5. 15 LoD4' de bina gösterimi: geometrik gösterim ve UML örnek diyagramı.....	44
Şekil 5. 16 LoD4'de CityGML bina modeli 3B grafik ve model hiyerarşisi.....	44
Şekil 6. 1 CityGRID'de bina modeli hiyerarşisi.....	51
Şekil 6. 2 CityGRID'de bina semantiği	52
Şekil 6. 3 Uygulama İşlem adımları	53
Şekil 6. 4 İstanbul Erkek Lisesi'nin Sketchup formatında 3B modeli.....	54
Şekil 6. 5 Çırağan Sarayı'nın Sketchup formatında 3B modeli	54
Şekil 6. 6 Atatürk Kitaplığı'nın Sketchup formatında 3B modeli	54
Şekil 6. 7 Ayasofya'nın Sketchup formatında 3B Modeli	55

Şekil 6. 8	İstanbul Erkek Lisesi için CityGRIDxml dönüşümü.	55
Şekil 6. 9	Atatürk Kitaplığı CityGRIDxml dosyası	56
Şekil 6. 10	Ayasofya CityGRIDxml dosyası	56
Şekil 6. 11	Çırağan Sarayı geometrik gruplandırma aşamaları.....	57
Şekil 6. 12	3B örnek modeller için CityGRID’de bina ve bileşenlerine ilişkin hiyerarşik semantik yapı	58
Şekil 6. 13	Çatı detayının (cephe) semantiği ve LoD3 temsili	58
Şekil 6. 14	Bina çatısının semantik yapısı ve LoD2 temsili.....	59
Şekil 6. 15	Doku giydirilmiş Çırağan Sarayı örneği.....	60
Şekil 6. 16	Atatürk Kitaplığı’nın LoD2 ve LoD3 ayrıntı düzeyinde temsili.....	60
Şekil 6. 17	Çırağan Sarayı’nın LoD2 (a) ve LoD3 (b) ayrıntı düzeyinde temsili.....	61
Şekil 6. 18	Atatürk Kitaplığı ve Çırağan Sarayı için CityGML öznitelik türleri	62

ÇİZELGE LİSTESİ

	Sayfa
Çizelge 4. 1	CityGML’de modelleme yapısı 24
Çizelge 4. 2	Tematik modüller ve tanımlı oldukları ayrıntı düzeylerini içeren ilgili detay türleri 25
Çizelge 5. 1	<i>AbstractBuilding</i> sınıfının semantik temaları 32
Çizelge 5. 2	CityGML’de ayrıntı düzeyleri için belirlenmiş çözünürlük ölçütleri 37
Çizelge 6. 1	CityGML’i destekleyen araçlar ve hizmetler..... 48

**BİNA NESNELERİNİN CITYGML STANDARDINA GÖRE LoD2 VE LoD3
AYRINTI DÜZEYLERİNDE ÜÇ BOYUTLU MODELLENMESİ**

Şüheda ÖZDOĞAN

Harita Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Doç. Dr. Ali Melih BAŞARANER

Kentlerin giderek büyüyen ve karmaşıklaşan yapısı, yönetim süreçlerine gelişkin bilgi ve iletişim teknolojilerinin dahil edilmesi gereksinimini artırmıştır. Buna paralel olarak, farklı alanlarda üç boyutlu (3B) kent modellerinin oluşturulması ve kullanımı yaygınlaşmaktadır. 3B kent modelleri, gürültü yayılma simülasyonu ve haritalaması, kent ve telekomünikasyon planlaması, afet yönetimi, eğitim amaçlı gerçek zamanlı simülasyonlar ve tesis yönetimi gibi çeşitli uygulamalarda mekansal analiz ve görselleştirme amaçlı kullanılabilir. Bu nedenle, yerel yönetimler 3B kent modellerini oluşturma çalışmalarına başlamıştır. Bu kapsamda, 3B CBS ve sanal coğrafi ortamlar gibi yeni kavram ve teknikler halen geliştirilme aşamasındadır.

3B kent modellerini temsil eden ortak bir standart olarak CityGML, farklı kaynaklardan gelen 3B kent verilerini depolamak, yönetmek ve entegre etmek için, Açık Mekan Konsorsiyumu (OGC) tarafından geliştirilmiştir. CityGML, 3B geometrileri, temaları ve semantikleri, öznitelikleri ve ilişkileri destekleyen ve gerçekçi görselleştirme için gelişmiş seçenekler sunan nesne yönelimli modeldir. CityGML bir kente ait bina, ulaşım ve bitki örtüsü gibi temel nesneleri tanımlamaktadır.

Bu 3B kent modelleri, çoklu ölçeklerde oluşturulmaktadır. CityGML standardı beş ayrıntı düzeyinde (LoD) mimari, kentsel ya da bölgesel kapsamda, amaca uygun birçok topografik nesnenin üç boyutlu geometrisi, topolojisi, semantiği ve görünüşünü (görsel temsili) tanımlar. 3B geometri ve görünüş bilgilerine ek olarak, yukarıda söz edilen uygulamalar özellikle kapsamlı semantik bilgiler gerektirirler. Modeller, daha düşük

doğruluklu 2.5B model (LoD0)'den başlayıp ve kapsamlı iç mekan modeli (LoD4)'nde sonlanan beş ayrıntı düzeyinde temsil edilmektedir.

Ülkemizde şimdiye kadar 3B kent modelleri daha çok belediyelerce görselleştirme amaçlı olarak oluşturulmuştur. Bu modeller ileri kentsel analiz süreçlerine, henüz etkin olarak dahil edilmiş değildir. Ayrıca şu ana kadar oluşturulan modeller, yaygın kabul görmüş standartları desteklememektedir. CityGML standardına uygun 3B kent modeli oluşturma çalışmaları, henüz çok yeni ve çok az sayıdadır.

Bu tez çalışmasında, CityGML standardı incelenmekte ve bina nesnelerinin farklı ayrıntı düzeylerinde modellenmesine ilişkin kavramlar ve teknikler irdelenmekte ve bu kapsamdaki gelişmeler ele alınmaktadır. Bu amaçla ilgili yazılımlardan bazıları incelenmiş ve CityGML standardında 3B kent modelleme için en uygun yazılım olan CityGRID™ tercih edilmiştir. Uygulamada İstanbul iline ait mevcut 3B bina verileri, CityGML standardına uygun olarak LoD2 ve LoD3 ayrıntı düzeylerinde modellenmiştir. Modelleme aşamaları kapsamlı olarak incelenmiş, uygulamada karşılaşılan sorunlar ortaya konmuş ve gelecekteki çalışmalara yönelik bazı öneriler ve değerlendirmeler yapılmıştır.

Anahtar Kelimeler: CityGML, Bina verileri, 3B Kent Modelleme, Ayrıntı Düzeyleri (LoD)

**THREE-DIMENSIONAL MODELLING OF BUILDING OBJECTS AT LoD2 AND
LoD3 IN ACCORDANCE WITH CITYGML STANDARD**

Şüheda ÖZDOĞAN

Department of Geomatic Engineering

MSc. Thesis

Adviser: Assoc. Prof. Dr. Ali Melih BAŞARANER

Owing to continuously growing and complicating structure of cities, it is highly necessary to include advanced information and communication technologies in their management processes. In parallel, the creation and the use of three-dimensional (3D) city models are widespreading in different areas. 3D city models can be used for spatial analysis and visualization in various applications such as noise propagation simulation and mapping, urban and telecommunication planning, disaster management, real-time simulations for training purposes, and facility management. Therefore, local governments have started to work on building 3D city models. In this context, new concepts and techniques such as 3D GIS and virtual geographic environments are still under development.

In order to store, manage, and integrate data from multiple sources, CityGML has been developed as a standard by Open Geospatial Consortium (OGC) to represent 3D city models. CityGML is an object-oriented model that supports 3D geometry, themes and semantics, and attributes and relationships as well as presents sophisticated options for realistic visualization. CityGML defines the main objects in a city, such as buildings, transportation, and vegetation.

These 3D city models have been constituted at multiple scales. CityGML standard describes 3D geometry, topology, semantics and appearance (visual representation) of many relevant topographic objects in urban or regional context at five levels of detail

(LoDs). In addition to 3D geometry and appearance information, above mentioned applications particularly require advanced semantic information. The model starting from 2.5D less accurate model (LoD0) and ending with very detailed indoor model (LoD4).

In our country, the 3D city models have been so far generated by municipalities mainly for visualization purposes. These models have not been effectively included in urban analysis processes yet. Besides, the available models have not supported the widely adopted standards. 3D city modeling works complying with CityGML standard are too new and few.

In this thesis study, CityGML standard is investigated and concepts and techniques for modeling building objects at different levels of detail are examined as well as the developments in this context are discussed. For this purpose, some of the related softwares have been examined and CityGRID™ was preferred as the most appropriate software among them. In practice, existing 3D building data belonging to the province of Istanbul has been modeled at LoD2 and LoD3. Modeling phase is investigated in detail and practical problems are presented as well as some suggestions and evaluations for future research are made.

Keywords: CityGML, Building Data, 3D City Modeling, Levels of Detail (LoDs)

1.1 Literatür Özeti

Sanal üç boyutlu (3B) kent modelleri; gürültü haritalama gibi çevresel simülasyonlar, eğitim simülatörleri, afet yönetimi, navigasyon, mimarlık ve kent planlama ile ilgili çok sayıda uygulama ile toplum içinde ve insanların günlük hayatında gittikçe artan bir rol oynamaktadır [1]. Bu nedenle bir çok kent ve şirket, hem manuel hem de otomatik olarak kendi 3B modellerini oluşturmaya başlamışlardır. 3B kent verilerinin zenginleştirilmesi ile birlikte, farklı kaynaklardan gelen verilerin entegre edilmesi ve paydaşlar, karar vericiler ve kamu için hızlı ve kolay bir şekilde elde edilebilir 3B kent sahneleri haline getirilmesi son derece önemlidir [2]. Günümüzde, çok sayıda etkileyici ve gerçekçi 3B kent modeli, düzenli olarak bilimsel, mesleki ve ticari etkinliklerde sunulmaktadır [3].

DXF, SHP, VRML, X3D, KML, COLLADA, IFC¹, CityGML ve 3D PDF gibi kabul görmüş ve uluslararası standartlar karşılaştırıldığında, her 3B standardın belirli amaçlar için tasarlandığı anlaşılmaktadır. COLLADA ve IFC, çok farklı türde geometrileri desteklemektedir. VRML, X3D ve COLLADA, gerçekçi dokuları desteklemede en ileri seviyededir. IFC haricinde tüm bu standartların, semantik ve öznitelik desteği zayıftır. Açıkçası, bu standartlar CAD alanından türemiştir. Bunun aksine, SHP, IFC ve CityGML gibi standartlar, semantik, nesneler, öznitelikler ve nesneler arasındaki ilişkiler hususunda çok güçlü desteğe sahiptirler. Bu nedenle, bu standartlar yalnızca

¹ Bina bilgi modelleme alanında yapı nesnelerini modellemede kullanılan en önemli standart.

görselleştirme değil, aynı zamanda mekansal analiz için de önemli olan bilgileri tutmaktadırlar [4], [5], [6].

CityGML ile, 3B kent modeli uygulaması için gerekli olan temel varlıklar, öznitelikler ve ilişkileri tanımlanmaktadır ve CityGML, sanal 3B kent modellerinin semantik, geometri, topoloji ve görünüş gibi dört farklı yönünü temsil etmektedir [5], [7], [8].

CityGML, veri alış-veriş formatı olarak tasarlanmış ve veri tabanı şeması sunan nesne yönelimli 3B bilgi modelidir. Model, biçimsel olarak UML ve XML kullanarak tanımlanmaktadır. Teknik olarak CityGML, GML uygulama şeması olarak tanımlanmıştır ve detayları tanımlamak için GML mekanizmalarını kullanır. Bu nedenle, coğrafi web servisleri aracılığıyla mekansal veri alış-verişi ve düzenlemesi gittikçe daha fazla önem kazanacağından, coğrafi referanslama ve web kullanımı desteği nedeniyle CityGML mekansal veri altyapıları kavramına mükemmel bir şekilde uymaktadır. Böylelikle başlangıçta yerüstü nesneleri için tasarlanan CityGML, uygulama alanı eklentisi (ADE) mekanizması aracılığıyla teknik altyapı tesisleri ve tüneller gibi yer altı nesnelerini entegre edebilmek ve özel uygulamalar için mevcut detay türlerine farklı detay türleri ve özellikleri eklemek için genişletilebilmektedir [3], [7], [9], [10].

Bir nesnenin tematik anlamı, CityGML'de kapsamlı olarak geliştirilmiştir. CityGML tematik modülü, bina (*Building*), arazi kullanımı (*LandUse*), bitki örtüsü nesneleri (*VegetationObject*), kent donatıları (*City Furniture*) gibi çeşitli tematik detay türlerini içermektedir [3], [7], [9].

CityGML standardının odak noktası, tüm detayların semantik olarak tanımlanmasıdır: Binalar ve parçaları (bina bölümleri, duvarlar, çatılar, çatı penceresi, kapılar, pencereler vb.), ulaşım nesneleri, su kütleleri, bitki örtüsü, arazi ve diğer kent detaylarıdır. Dahası, bu detaylar arasındaki ilişkiler (örn. Duvar ve kapı ilişkisi) açık bir şekilde temsil edilmektedir [5], [7].

Geometri (ve topolojiyi) temsil etmek için CityGML, coğrafya işaretleme dili (GML)'nce sağlanan standartlaştırılmış bir model kullanır. Bu da geometrik temsil düzeyinde veri bütünleştirmeyi kolaylaştırır [4], [5], [7].

CityGML, ayrıntı düzeyleri biçiminde geometrinin çok çözünürlüklü sürdürülebilirliğini sağlamaktadır. Ayrıntı düzeyleri OGC tarafından beş ayrıntı düzeyinde açıklanmaktadır:

LoD0, özünde hava fotoğrafı veya topografik haritalarla kaplanan 2.5B sayısal arazi modeli (SAM)'dir. LoD1, düz çatılı prizmatik binaları kapsayan blok modeldir [3], [11]. LoD2'de ise bir bina farklılaştırılmış çatı yapılarına ve tematik olarak farklılaştırılmış yüzeylere sahiptir. LoD3, kapsamlı duvar ve çatı yapıları, balkonlar, girinti ve çıkıntılar ile mimari modelleri temsil etmektedir. Yüksek çözünürlüklü dokular, bu yapılar üzerine kaplanabilir. LoD4, LoD3'e iç yapılar eklenerek oluşturulmaktadır. LoD0, daha çok düşük doğruluğa sahip bölgesel model olarak yapılandırılmıştır. Öte yandan LoD4, en kapsamlı ve aynı zamanda en yüksek doğruluklu modeldir. Ayrıntı düzeyleri, genellikle sadece geometrik temsil olarak anlaşılmaktadır. Aslında belirli düzeylerde geometrilere bir anlam atanmıştır. Örneğin, Çatı Yüzeyi (*RoofSurface*) için LoD2'de bir temsil biçimi uygulamaya konmaktadır. Ancak, CityGML spesifikasyonları, ayrıntı düzeylerinin oluşturulması için katı kurallar dayatmayıp semantik ve geometri arasındaki tutarlılığa karar verme özgürlüğünü kullanıcıya bırakmaktadır [3], [7], [8].

Tüm ayrıntı düzeylerinde gerçekçi görselleştirmeye yönelik olarak dokular eklenebilmesi için, nesnelerin çoğu yüzeyler ile temsil edilmektedir. CityGML görünüş (*appearance*) modeli, renklendirme ve dokulandırma için yönergeler sağlamaktadır [5], [12].

Bina modeli, CityGML'in en kapsamlı tematik kavramlarından birisidir. Binalar, beş farklı ayrıntı düzeyinde (LoD0-LoD4) temsil edilmektedir. En düşük düzey olan LoD0, detayların (topografik veri tabanında yer alan) 2.5B (2B geometri + yükseklik özneteliği) temsildir. LoD1, düz çatılardan oluşan blok modellerini içerir. Bu blok modelleri dokularla eşleştirilmiş olabilir. LoD2, belirli çatı yapıları ve tematik olarak farklılaştırılmış yüzeyler ile binaları tanımlar. LoD3, ayrıntılı duvar, çatı yapıları ve balkonlar vb. ile modelleri tanımlar. Dokular ayrıca bu yapılar üzerine eşleştirilebilir. LoD4, LoD3'teki 3B nesnelere iç yapı eklenerek oluşturulur [5], [9], [12], [13].

Bu alanda ülkemizde yapılan çalışmalar incelendiğinde, Yücel (2009) doktora tezi kapsamında, 3B kent modelleme için ayrıntı düzeyleri kavramını incelemiş ve seçilen bir uygulama bölgesine yönelik dört farklı ayrıntı düzeyinde 3B kent modeli oluşturmuştur. Çalışmada daha çok 3B geometrilerin ve görünüşlerin modellenmesine odaklanılmıştır. Ülkemizde kullanılmakta olan mevcut verilerin bu konudaki yeterliliğini araştırmış ve eksik yanlarını ortaya koymuştur [14]. Şengül (2010) yüksek lisans tezi

kapsamında fotogrametrik yöntemler kullanılarak oluşturulan 3B bina modeli ile CityGML bina veri modelinin bağdaştırılabilirliğini incelemiş ve her iki model arasındaki benzerlik ve farklılıkları belirlemiştir [15]. Karaş ve Ekmekçi (2011) CityGML modeli için gerekli semantik modelleme ve LoD kavramları üzerinde durmuş, oluşturulan modelin geometrik iyileştirilmesi ve doku kaplamanın CityGML'deki uygulamasını açıklamıştır [16].

1.2 Tezin Amacı

Bu tez çalışmasında birinci amaç, öncelikli olarak CityGML standardının özellikle 3B binalara yönelik sunduğu modelleme olanaklarının incelenmesidir. İkinci amaç, farklı formatlarda modellenmiş ve doku giydirilmiş 3B binaların, LoD2 ve LoD3 ayrıntı düzeyine uygun olarak modellenmesidir. Üçüncü amaç ise, bu 3B modellerin CityGML standardına dönüştürülebilirliğinin test edilmesidir.

1.3 Hipotez

Bu çalışmada hipotez aşağıdaki gibi kurulmuştur:

- Belirli bir standarda sahip olmayan 3B kent modelleri, binalar kapsamında yalnızca görsel değil semantik ve geometrik yapısıyla uluslararası kabul görmüş CityGML standardında modellenebilir.
- 3B bina modelleri, farklı ayrıntı düzeyleri biçiminde yapılandırılabilir.
- Ülkemizdeki 3B kent modelleme çalışmalarına katkı sağlanabilir ve/veya yön verilebilir.

3B KENT MODELLEME

2.1 3B Mekansal Modellerin Temsili İçin Veri Yapıları

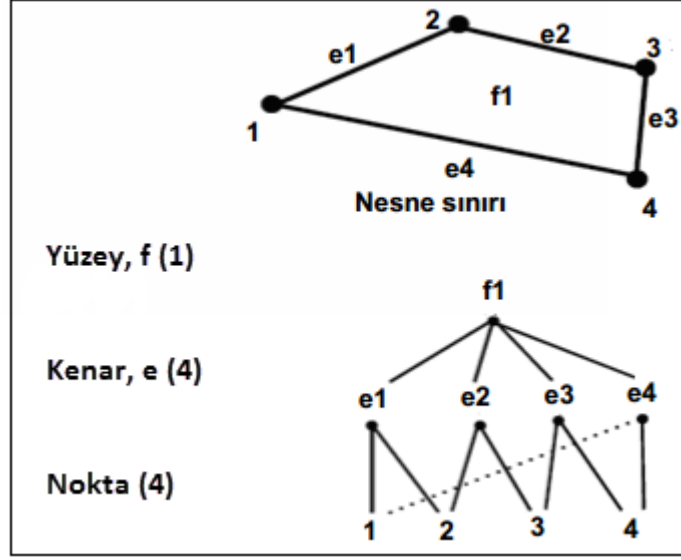
Nesne temsilleri, yüzey tabanlı ve hacim tabanlı olarak sınıflandırılabilir. Nesne, eğer yüzey ilkellerince (*primitive* - alt öge) temsil ediliyorsa yüzey tabanlı modelleme, eğer nesnenin iç yapısı, katı (*solid*) bilgileri ile tanımlanıyorsa hacim tabanlı modelleme söz konusudur. Yüzey tabanlı modeller grid, şekil modeli (*shape model*), yüzey modeli (*facet model*) ve sınır temsildir (*boundary representation*). Hacim tabanlı modeller de 3B dizi (*Voksel*), sekizli ağaç yapısı (*Octree*), yapısal katı geometri (*Constructive Solid Geometry*) ve 3B Düzensiz Üçgen Ağ (TIN)'dır [17],[18].

Bu bölümde binalar için yaygın olarak kullanılan sınır temsili, yapısal katı geometri ve parametrik model gibi 3B mekansal veri yapıları açıklanmaktadır [19].

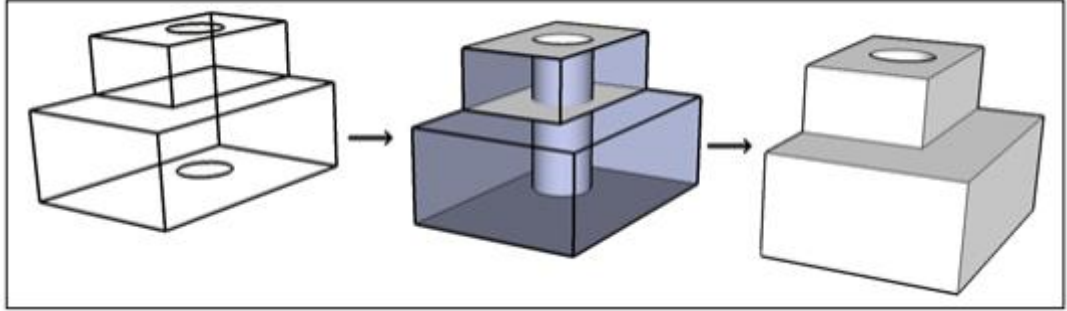
2.1.1 Sınır Temsili (B-Rep) Modeli

Bir nesneyi önceden tanımlanmış nokta, kenar, yüz ve hacim ilkellerinin kombinasyonu ile ifade eder. Nokta ögelerinin örnekleri; tek noktalar, yükseklik eğrisi noktaları ve diğer yardımcı noktalardır. Kenar ögelerinin örnekleri; düz çizgiler, yaylar ve çemberlerdir. Yüz ögelerinin örnekleri; çokgen düzlemler, koni ve silindir yüzleri gibi diğer mekansal nesne yüzleridir [18]. 3B koordinat (x,y,z) ile noktalar çizgileri; çizgiler yüzeyleri ve yüzeyler de ana blokları oluşturmaktadır [20], [21]. Hacimler, B-Rep'te hacim karakteristiklerini temsil eden yüzey elemanlarının bir uzantısıdır. Bloklar; prizma, koni, silindir veya bu üçünün birleşimi olabilir. Bu nesneler, sonrasında 3B

nesne oluşturmak için biraraya gelirler. Ardından bu nesnelere renk veya doku atanır. Kendi sınırları ile 3B kent nesnelerinin temsili, 3B CBS'de en başarılı veri modelidir ve CityGML gibi projelerde benimsenmektedir. Bu yöntem, uzunca bir süredir 3B grafiğin temelini oluşturmaktadır [21]. Tam olarak temsil edilen topoloji, nesnelerin kolay bir şekilde bakımını ve güncellenmesini sağlamaktadır fakat karmaşık yapılar hesaplama yükünü artırmaktadır [20].



Şekil 2. 1 B-Rep'e ilişkin düzlemsel çokgen temsili [14]



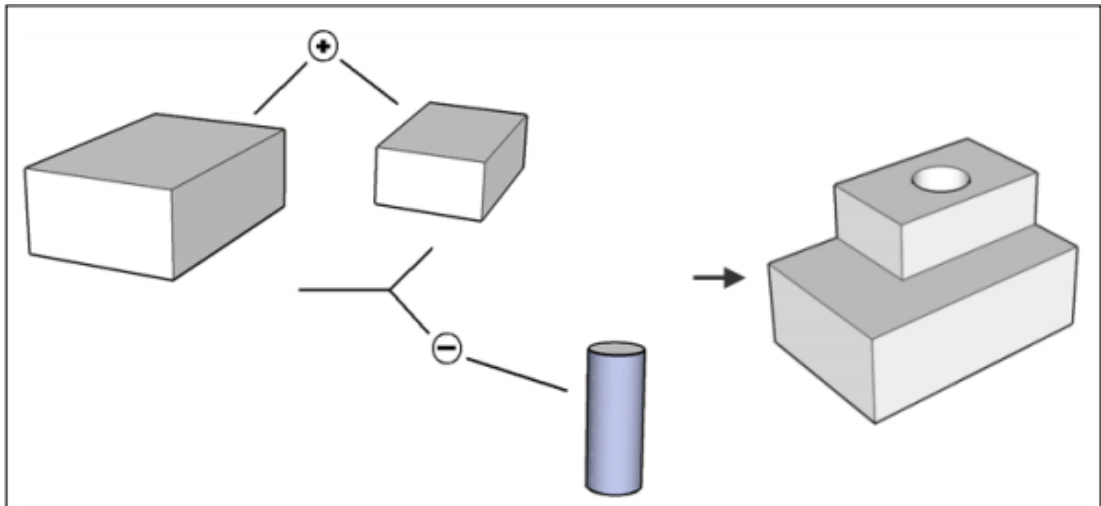
Şekil 2. 2 3B bir nesnenin 2B yüzeyler ile B-Rep temsili [14]

Bu model ile bir nesneyi temsil etmek için, nesnenin sınır temsili (*B-Rep*) ögesi, geometrik veri ögesi, ögenin tanımlama kodu ve diğer ögelerle ilişkisi olması gerekir. Şekil 2.1, bir çokgen nesnenin basit bir B-Rep temsili vermektedir. Burada, nesne yapısının temel özelliği, ilkelerin kombinasyonu olmasıdır. Yani noktaların kombinasyonu kenar, kenarların kombinasyonu düzlem yüzey oluşturur [18]. Şekil 2.2'de ise bir 3B nesnenin 2B yüzeyler ile modellenmesi görülmektedir [14].

Düzlemsel olmayan yüzeylerin B-Rep temsiliinde çok fazla geometrik ve karmaşık hesaplama gereklidir. B-Rep, bilgisayar destekli tasarım ve üretimde (CAD/CAM) tercih edilmesine rağmen, hesaplama karmaşıklığı ve etkisiz mantıksal (Boolean) işlemleri nedeniyle yalnızca düzenli ve düzlemsel nesneler için uygundur [14].

2.1.2 Yapısal Katı Geometrisi (CSG)

Yapısal katı geometri; küp, silindir ve küre gibi önceden tanımlanmış temel öğeleri birleştirerek üç boyutlu bir dünya oluşturur (Şekil 2.3). Farklı nesneler, temel öğeler dönüştürülerek (örn. yeniden boyutlandırma) oluşturulur. Sanal Gerçeklik Modelleme Dili (VRML) tarafından bu yaklaşım kullanılmaktadır. Basit veri yapıları ve düşük veri hacimleri, işlem sürelerinde azalmaya yol açar. Ancak topoloji tam olarak tutarlı değildir ve az sayıda temel öğe, mekansal nesnelerin temsili için mevcuttur [20].



Şekil 2. 3 CSG ile modellenmiş 3B nesne [14]

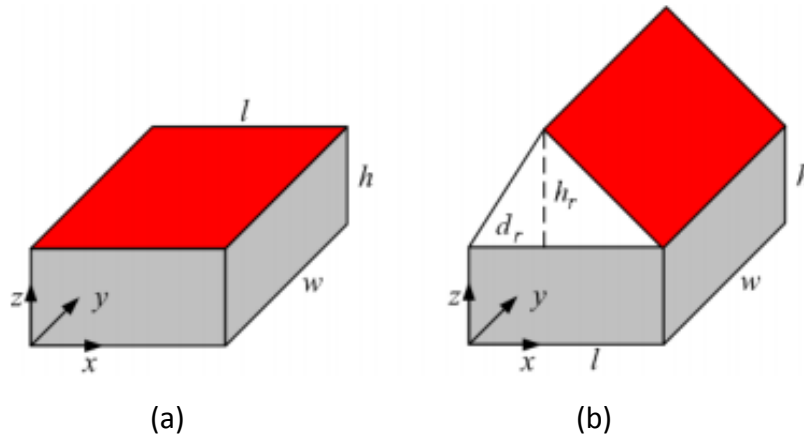
Nesne oluşturma, bir temel modelleme dili ile etkileşimli bir şekilde tamamlanabildiği için yapısal katı geometri, CAD/CAM gibi katı modelleme uygulamalarında yaygın olarak kullanılmaktadır. Bu temsil, ilkeleri ya da katı geometrileri oluşturmak genellikle sorunsuz olduğu için aynı zamanda mühendislikte ve mimari görselleştirmelerde de çok tercih edilmektedir [18].

Yapısal katı geometri ilkeleri, düzenli şekilli hacimsel yapı türlerindendir ve geometrik dönüşüm ve *Boolean* işlemleri kullanılarak kombine edilmektedir. Geometrik dönüşümler, öteleme, döndürme ve ölçeklendirmeyi gerektirirken, Boolean işlemleri de birleşim, kesişim ve çıkarmayı (farkını almayı) gerektirir. İlkelerin sayısı arttıkça,

yapısal katı geometrisi modelinin depolama yüzeyi de artar. Önceki araştırmalar, yapısal katı geometrisi modelinin yalnızca düzenli yapıdaki nesneleri tanımlamak için uygun olduğunu göstermiştir. Çünkü düzensiz şekilli hacimsel örnekleri oluşturmak için düzenli şekle sahip nesnelerin ilkel kombinasyonları dikkate değer hesaplama yükü getirir. Bu nedenle, şu aşamada yapısal katı geometrisi modelinin düzensiz nesneler için pek uygun olmadığı düşünülmektedir [14], [18].

2.1.3 Parametrik Model

Parametrik model, bina ilkellerinin geometrisini temsil etmek için şekil parametrelerini; bina ilkellerinin ilişkilerini belirlemek için duruş parametrelerini kullanmaktadır. Bu durum, bir bina sınıfının yapısal özellikleri hakkındaki bazı ön bilgilere dayanmaktadır. Örneğin, düz çatı yapısının geometrik özellikleri Şekil 2.4a'da gösterildiği gibi bir dikdörtgensel hacim tarafından kodlanabilir. Bu türdeki binaları tanımlamak için yedi tane parametre gerekmektedir: Şekil parametreleri, genişlik (w), uzunluk (l), yükseklik (h), duruş parametreleri ise xy -düzleminde mekansal konum ve dönüklük için koordinatlardır (x , y , z). Bir başka örnek, bir dikdörtgensel hacim ve bir üçgensel hacimden oluşturulabilen üçgen çatılı binalar olabilir (Şekil 2.4b) [15].



Şekil 2. 4 Parametrik Bina Modeli: (a) düz çatılı bina, (b) üçgen çatılı bina [15]

Düz çatılı binalar ile karşılaştırıldığında, üçgen çatıyı tanımlamak için iki tane ekstra parametre eklenmek zorundadır: çatı sırtının yüksekliği h_r ve çatı referans noktasından sırt taban noktasına olan mesafe d_r . Benzer şekilde, L-şeklinde bir yapı tanımlanabilir. Ancak kapsam, parametre sayısının önsel belirlenmesi nedeniyle sınırlıdır [19].

2.2 Yaygın 3B Mekansal Veri Toplama Yöntemleri

3B kent modeli oluşturulurken, sonuç modelde hangi detayların dikkate alınması gerektiğini bilmek önemlidir. Kent modeli için temel detaylar, modele doku bilgisi eklemek için kullanılan ortofotonun yanı sıra bölgenin topografik modelidir [22]. Coğrafi veri toplamak için çok sayıda yöntem mevcuttur. Bunlar; Global Navigasyon Uydu Sistemleri (GNSS) ve jeodezik ölçmeler, lazer tarama, fotogrametri ve uzaktan algılama ve diğer teknolojik yöntemlerdir [23], [24]. Fotogrametrik yöntem, kamera iç dönüklük elemanları; konum ve kamera dönüklüğü; görüntü ölçümleri ve yer kontrol verilerini esas alarak noktanın nesne uzay (3B) koordinatlarını elde etmek için matematiksel hesaplamalar gerçekleştirerek 2B görüntüden 3B bilgi sağlamak amacıyla yaygın olarak kullanılmaktadır [22].

Bir noktanın (bindirilmiş görüntülerde) birçok kez ölçüldüğü fotogrametrinin aksine, LiDAR Sisteminde, nesne uzay noktasını bir kez ölçülmektedir. LiDAR sistemi, lazer tarama, GPS alıcıları ve inersiyal navigasyon sistemini içerir. LiDAR sistemi, lazer sinyali yayarak ve sinyalin dönüş süresini ölçerek çalışmaktadır. Sinyalin dönüş süresine dayalı olarak, tarayıcı ve ölçülen nokta arasındaki mesafe bulunmakta ve nesne koordinatlarını hesaplamak için GPS ve INS gözlemleri ile birlikte kullanılmaktadır.

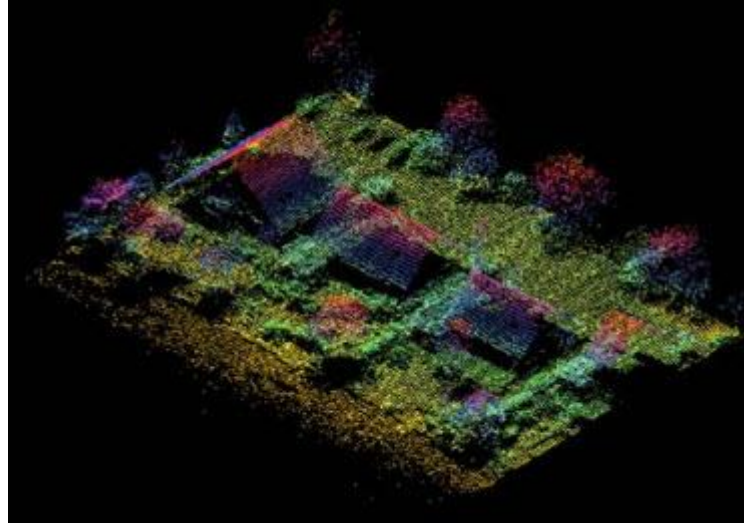
3B öznitelik bilgilerini elde etmek için üçüncü bir yöntem, arazi ölçmeleridir. Ölçme teknikleri, konum bilgilerini elde etmenin en geleneksel biçimidir. “Total Station” ve/veya GPS ölçme tekniklerinin kullanımını içeren çeşitli ölçme teknikleri vardır. Bu belirtilen yöntemlere ek olarak veri, mevcut veri tabanı içinde bulunan verilerden ve haritaların sayısallaştırılmasıyla da elde edilebilmektedir [22].

2.2.1 Lazer Tarama (LiDAR)

Lazer tarama, yüzey üzerinde bulunan detayları hızlı ve ayrıntılı bir şekilde nokta bulutu olarak adlandırılan nokta kümesi biçiminde mekansal veri toplama teknolojisidir. Lazer tarayıcılar, yersel platformların (genellikle sehpalar) yanı sıra havada taşınır platformlarda da çalıştırılır [25].

Lazer tarama teknolojileri veya yüksek tanımlamalı ölçüm (*high definition survey*) ile fiziksel yapı veya detayların doğru 3B sayısal temsilini otomatik olarak oluşturmak mümkündür. Yaygın kullanılan LiDAR teknolojisi, nesne mesafesini belirlemek için kısa

lazer sinyalinin gidiş-dönüş süresini hesaplamaktadır. Sonuç, 3B görselleştirme sistemi içerisine dahil edilmeye uygun hale dönüştürülebilen nokta bulutudur. Şekil 2.5'te 15 nokta/m² yoğunluklu nokta bulutu gösterilmektedir [25].



Şekil 2. 5 Lazer tarama - nokta bulutu [25].

2.2.2 Fotogrametrik Ölçme Yöntemleri

Fotogrametri teknolojisi kullanılarak elde edilen veriler yardımıyla 3B kent modelleme ve kentsel görselleştirme, sayısal mimaride en ilgi gören araştırma konulardan biridir [15]. Fotogrametri yüzeylerden ve nesnelerden 3B ölçmeler elde etmek için fotoğraf kullanan bilimsel uygulama alanıdır ve 3B kent modellemede en çok tercih edilen teknolojiler arasında yer almaktadır. Bu yöntem kullanılarak çoklu görüntülerden nesnenin özellikleri otomatik olarak çıkartılmaktadır [26], [27]. Kent hakkında mekansal bilgi edinebilmek için fotogrametride iki temel yaklaşım kullanılmaktadır [15]. Bunlardan birincisi, sayısal kamera ile yersel fotoğraf çekmektir. Bu yaklaşımın avantajı, fazla maliyetli olmamasıdır. Öte yandan, fotoğraflar binanın her cephesinden alınmalıdır ancak bazen güvenlik ve gizlilik hususlarından dolayı fotoğraf çekmek zorlaşmaktadır [15], [27]. Yöntemin diğer dezavantajı, fotoğraf eşlemede doğru modeli bulmak için daha fazla zaman harcanmasıdır [15]. Ayrıca her binaya ait tüm yükseklikleri içeren bir kaç adet fotoğraf çekmek gerektiğinden çok sayıda görüntü dosyası yönetmek gerekir. Bunun yanında, farklı beyaz dengelere (*white balances*) sahip aynı binanın iki görüntüsünü eşleştirmek zordur.

Diğer yöntem de, havadan ya da uydudan alınmış görüntülerdir. Avantajlı olmasının nedeni, az sayıda görüntü kullanmanın yeterli olmasıdır. Aynı görüntüden çıkarılan binaların dokusu, görüntünün renk dengelerini düzeltmek için gerekli değildir fakat, yersel fotoğrafları çekmekten daha maliyetlidir [27]. Ayrıca, havadan alım durumunda uçuş planını ve iç ve dış yöneltme parametrelerini ayarlamak gerekmektedir [15].

2.2.3 GNSS Verileri

GNSS kullanıcıya kolay, hızlı ve ekonomik olarak konum belirlemesine izin veren bir diğer veri elde etme yöntemidir [28]. GNSS, uydulardan ve kontrol sistemlerinden oluşan ve günün herhangi bir anında yeryüzüne ait konum bilgilerinin elde edilmesini sağlayan bir sistemdir. Bu alanda Amerika'nın NAVSTAR GPS, Rusya'nın GLONASS, Avrupa'nın ise 2008 yılında kullanılmaya başlayan GALILEO adında sistemleri vardır. Amerikan sistemine ait 24 adet uydu dünya etrafında kendi yörüngelerinde dolaşmaktadır. Bu uyduların yörünge süreleri 12 saattir ve yörüngeleri sürekli gözlenerek konumları hassas bir şekilde belirlenmektedir. Uydular ve yeryüzündeki algılama istasyonları benzer radyo dalgası ve kodlama sistemi kullanır. Radyo dalgalarının algılama istasyonu tarafından alınması ve veri dönüşüm sırasında bir gecikme meydana gelmektedir. Bu gecikme miktarı, uydu ve algılama istasyonu arasındaki uzaklığa bağlıdır. GNSS kullanıcıları bulundukları konumun 3B koordinatlarını (x, y, z), kullandıkları GNSS aletinin kapasitesine göre farklı doğrulukta elde ederler [14].

2.3 3B Kent Modellerinin Kullanım Alanları

Yaşam kalitesinin gelişmesi ile birlikte, kentlerde yaşayan insan sayısı giderek artmaktadır. Bu nedenle, modern kentlerin yönetimini desteklemek için etkin teknolojiler geliştirmek önemlidir. 3B kent modellerine ilişkin teknolojiler, ilgili çok sayıda topluluk tarafından talep görmektedir [29].

3B kent modelleri, kentsel planlama, trafik kontrolü, afet yönetimi, mobil telekomünikasyon tasarımı gibi kent yönetimi ile ilgili uygulama alanlarında geniş çaplı kullanılmaktadır. Bu nedenle pek çok ülke ve kent, resmi 3B modellerini oluşturmakta ve erişime açmaktadır. Örneğin Berlin'in 3B kent modeli, Mart 2009'dan bu yana halka

açıktır. Resmi Berlin Modeli, 892 km² alanı kapsayan toplamda dokulandırılmış 474 000 binayı içermektedir (Şekil 2. 6). Bu model, Google Earth aracılığıyla görüntülenebilir ve CityGML formatında mevcuttur [29], [30].



Şekil 2.6 Berlin 3B kent modeli [31]

Sanal 3B kent modelleri, afet yönetimi için önemli bilgiler sağlamaktadır. Kent modelleri, öncelikle kentin şekli ve yapısı hakkında bilgi vermektedir. Deprem gibi felaketlerden dolayı altyapının zarar görmesi durumunda, bu referans verilere anında erişim, hasarın boyutunun hızlı bir şekilde değerlendirilmesi ve zarar gören alanların yeniden onarılması olanağı tanımaktadır. İkinci olarak, 3B kent modelleri, iç ve dış mekan navigasyonuna yönelik 3B görselleştirmeler ve yer belirleme olanağı sağlamaktadır. Artırılmış gerçeklik sistemleri, gerçek dünya görünümüyle görsel olarak çakışan bilgiler vermektedir. Bu sistemler, çakıştırılan grafiklerin konumlarını ve kapatmalarını hesaplamak amacıyla 3B kent modellerine gereksinim duyarlar. Üçüncü olarak, binaların içinde ve dışındaki 3B tahliye yolları, uygun kent modeliyle belirlenebilir. Dördüncü olarak, taşkın senaryosunda 3B kent modelleriyle etkilenen bina katları belirlenebilmektedir [10].

2.3.1 3B Kent Modeli Oluşturma ve Yönetimi

3B modellerin çoğu manuel olarak oluşturulmaktadır. 3B modelleme araçları artmalarına rağmen, hala büyük kentsel alanları modelleme ihtiyacını karşılayamamaktadırlar. Günümüzde birçok kent, dokuyla kaplı 3B kent modellerini hızlıca oluşturmak için LiDAR, sayısal kamera (mobil veri toplama) ve fotogrametri

teknikleri kullanmaktadır. Üç boyutlu kent modelleri, mevcut yapı bilgi modelleme (BIM) veya CAD dosyalarından da dönüştürülebilmektedir. 3B kent modelleri farklı kaynaklardan geldiği için 3B kent modellerinin tanımlanması, depolanması ve alış-verişi için bir standart gerekli olmuştur. Bu amaçla, 2008 yılında Açık Mekan Konsorsiyumu (OGC) tarafından, 3B kent nesnelerinin temsili için ortak bir bilgi modeli olarak CityGML yayınlanmıştır. CityGML, hem geometrik hem de semantik bilgileri içerir. Büyük miktardaki 3B kent modeli verisi nedeniyle Oracle ve PostgreSQL gibi veri tabanları, 3B kent modeli verilerini depolamak ve yönetmek için kullanılmaktadır [29].

2.3.2 3B Kent Modeli Görselleştirme

Görselleştirme, 3B kent ortamında karmaşık ve önemli bir konudur. Kentsel planlama ve afet yönetiminde kullanılan haritalar gibi çoğu 3B kent modeli uygulamasında görselleştirme kritik bir husustur. Farklı uygulamalara ilişkin genel görselleştirme gereksinimlerinden bazıları aşağıda verilmektedir:

Farklı ayrıntı düzeylerinde görselleştirme. Genellikle kent ortamını farklı ölçeklerde, (örn. bölgesel genel bakış ölçeğinden bina ya da oda gibi ayrıntılı ölçeğe kadar) görselleştirmeye gereksinim duyulur. Ayrıntı düzeyleri, farklı uygulama gereksinimlerine yönelik veri toplama süreçlerini ortaya koymak amacıyla farklı ölçeklerde modelleri temsil etmek ve veri analizi ve etkin görselleştirmeye olanak tanımak için önerilmiştir.

Zaman içindeki değişimlerin görselleştirilmesi. Değişimleri ortaya çıkarmak için kent modellerinin farklı zamanlardaki durumunu görmek gerekmektedir. Her nesnenin zamansal bilgisi, kentsel planlama gibi bazı uygulamalarda çok yararlı olabilecek 4B kent modeli oluşturmak için depolanmaktadır [29].

2.4 3B Modellemede Kullanılan Standartlar

3B kent modelleri için, X3D, VRML, 3D PDF veya 3ds Max/3D Studio MAX gibi 3B bilgisayar grafik formatları ya da KML ve COLLADA gibi coğrafi görselleştirme formatları kullanılmaktadır [7]. Bu formatlar 3B kent modellerini sunmak için kullanılabilir, fakat standartları temsil etmek için kullanılamamaktadır. Bu nedenle, farklı

standartlardan gelen verileri entegre etmek ve eksik semantik bilgilerden dolayı mekansal analiz yapmak zordur [6].

CityGML ve X3D entegrasyonu ile mekansal verileri görselleştirmek mümkündür. 3B kent verilerini CityGML’de temsil etmek ve X3D veya diğer 3B görselleştirme standartlarıyla görselleştirmek mantıklı bir yaklaşımdır [6].

CityGML, X3D ya da KML gibi görselleştirme standartlarının tamamlayıcısıdır. Bu görselleştirmeler, 3B modellerin sunumu, davranışı ve etkileşimine yönelik iken CityGML, 3B nesnelerin temelini oluşturan kent bilgilerinin değişimine odaklanmaktadır. Bu nedenle, özel amaçlı 3B görselleştirme teknikleri, 3B kent modellerinin daha etkin sunumu için gereklidir. KML, 3B model sunumları için COLLADA standardını kullanır. Coğrafi bilgileri saklamak ve Google Earth ve Google Maps gibi belirli istemcilerde coğrafi verileri görüntülemek için tasarlanmıştır [2].

X3D, coğrafi görselleştirme uygulamalarında önemli bir rol oynamaktadır. X3D ve KML/COLLADA karşılaştırıldığında, bu standartlar farklı şirketler tarafından belirli uygulamalar için tasarlanmaktadır; ancak OGC veya ISO tarafından onaylanmış uluslararası standartlar değildir. Bununla birlikte, 3B PDF dosyaları oluşturmak için U3D¹ formatında 3B kent modellerinin görselleştirilmesi gibi belirli amaçlar için kullanılabilir [29].

VRML, 3B tarayıcı eklentileri tarafından yaygın olarak desteklenmektedir ve sıkıştırılmış VRML kodu, X3D, 3Ds ve *Viewpoint* gibi diğer sıkıştırılmış kodlardan daha etkilidir [32]. Binalar için 3B sınır temsillerinin (*B-Rep*) en popüler şekli, yeri X3D tarafından alınan VRML’dir. VRML, örneğin bir binanın 3B geometrik tanımlaması için köşe noktalarının ve kenarlarının yanında, yüzey renkleri, UV² eşleştirmeli dokular, parlaklık, şeffaflık vb.’nin de belirtildiği bir metin dosyası formatı tanımlar [19].

CityGML, X3D, VRML veya COLLADA gibi 3B bilgisayar grafikleri standartları ve KML gibi coğrafi görselleştirme standartlarının tamamlayıcısıdır. Bu standartlar, 3B modellerin etkili ve nitelikli görselleştirilmesi ve etkileşimi için yöntemler ortaya koyar. Hem X3D hem de KML’de coğrafi referanslı koordinatlar kullanılabilir. X3D veya KML kullanarak

¹ 3B bilgisayar grafiği verileri için sıkıştırılmış dosya formatı standardı.

² UV eşleme, 3B bir modelin 2B görüntü temsili elde etmek için kullanılan 3B modelleme süreci.

semantik bilgi deęiřimi mmkn olmasına raęmen, her iki spesifikasyonda karmařık coęrafi detaylar ve onların iliřkilerinin nasıl temsil edileceęi standartlařtırılmamıřtır [5].

CityGML, X3D ve KML'nin kolayca tretilbildięi zengin kaynaklı bir format olarak kabul edilmelidir. CityGML, etkin grselleřtirme iin optimize edilmemiřtir. Ancak, CityGML nesnelerinin bina, kapı, bitki rts gibi tematik sınıflarla aıka iliřkilendirilmesiyle ortaya ıkan semantik bilgilerden nesneleri filtreleme ve 3B grafik řekiller, grnř zellikleri ve materyaller oluřturmakta da yararlanılabilir [5].

KML, XML ile coęrafi bilgileri temsil eden bir dil řemasıdır. KML, orijinal olarak Keyhole adlı bir řirket tarafından Goole Earth Viewer ile kullanılmak zere ilk olarak Keyhole Earth Viewer adıyla geliřtirilmiřtir. Bir KML dosyası, okgenler, grntler, 3B modeller ve yer iřaretleri gibi birok farklı detay ierebilir. Konumlandırma iin boylam, enlem ve ykseklięi ieren 3B coęrafi koordinatları kullanır [17].

2.5 3B Mekansal Modellemeye İliřkin Genel Problemler

Mevcut 3B kent modelleri ve kentsel uygulamaların sayısının artmasına raęmen, bu modellerin entegrasyonunu zorlařtıran bir takım yaygın problemler vardır [26]. 3B CBS oluřturulması, gerekli bileřenlerin ve farklı nesne trlerinin entegrasyonunu dikkate alırken gereklięi gsteren mekansal model ile ilgili ařaęıdaki problemlerinin zm gerekmektedir [18]:

- Farklı kuruluřlar: 3B kent modelleri, farklı řirketler, yetkili makamlar, niversiteler veya kentler tarafından retilmektedir ve herkes kendi ilgi alanına sahip olduęunda entegrasyon gleřmektedir. oęu durumda, hedeflerini kontrol edecek ve yol gsterecek sınırlı bteye sahiptirler [33].
- Farklı řemalar: retim ve geliřtirme sistemlerinde farklılıklara gre bu modeller, geliřtirilen modeller ve uygulamalar ile karřılařtırıldıęında entegrasyonu daha zor olan farklı format ve řemalara sahiptir.
- Farklı geometrik modeller: Bu modellerin kapsamı, hedefleri ve kořullarındaki farklılıklar dikkate alındıęında, uygulama gereksinimlerine (grselleřtirme, sorgulamalar, analizler vb.) gre geometrilerinde farklılıklar ortaya ıkmaktadır [33].

- Semantik kavramların eksikliği: Geometrideki farklılıklara göre, bu modeller ve uygulamalarına görselleştirme sürecinde gerek duyulmadığından semantik tanımların/kavramların eksik olduğu (özellikle görselleştirme modelleri) bazı durumları ifade eder.
- Birlikte çalışılabilirlikteki eksiklik: Modeller farklı dillerde geliştirilmektedir ve formatlarda, protokollerde, biçimlendirmelerde, kavramlarda ve semantiklerdeki belirgin farklılıkları açıkça gösteren araçlar kullanmaktadırlar. Modellere tek biçimli erişimi sağlamak da güçtür [33].

CityGML STANDARDI

3.1 CityGML' e Genel Bakış

CityGML, sanal 3B kent modellerinin depolanması ve alış-verişi için XML tabanlı bir formata sahip ücretsiz olarak kullanılabilen açık bir veri modelidir. Aynı zamanda Coğrafi İşaretleme Dili (GML) 3.1.1 versiyonu için uygulama şemasıdır [5], OGC ve ISO TC211 tarafından yayınlanan mekansal veri alış-verişi için genişletilebilen uluslararası standarttır [9], [34], [35]. CityGML, GML tabanlı olduğu için Web Detay Servisi (*Web Feature Service*), Web İşleme Servisi (*Web Processing Service*) ve Katalog Servisi (*Catalog Service*) gibi veri erişimi, işleme ve kataloglama için GML uyumlu tüm OGC web servisleri ailesiyle kullanılabilir.

CityGML'in geliştirilmesindeki amaç, 3B kent modelinin temel varlıkları, öznitelikleri ve ilişkilerinin ortak bir tanımlamasına ulaşmaktır. Bu, özellikle farklı uygulama alanlarında aynı verinin yeniden kullanımına olanak tanıyarak, 3B kent modellerinin uygun maliyetli sürdürülebilir bakımı açısından önemlidir [9].

CityGML, kent modellerinin geometrik temsili ve görünüşleri dışında, sayısal arazi modellerinin, bina, köprü ve tünelleri içeren alanların, bitki örtüsünün, su kütlelerinin, ulaşım tesislerinin ve kent donatılarının özellikle semantik temsillerini, tematik özelliklerini, taksonomilerini (aşamalı sınıflandırma) ve birleştirme hiyerarşilerini dikkate almaktadır. Farklı ayrıntı düzeylerini temsil yeteneği, CityGML'i küçük ve büyük alanlarda kullanıma uygun hale getirir. Altta yatan model, artan ayrıntı düzeyinde geometrik ve semantik açıdan daha ayrıntılı hale gelen beş farklı ardışık ayrıntı

düzeyinden oluşmaktadır [5].

CityGML, veri ile ilişkili semantikler ile birlikte hem karmaşık hem de coğrafi referanslı vektör verileri modeller. Diğer 3B vektör formatlarının aksine, CityGML, geometri ve görünüş bilgilerinin yanı sıra zengin genel amaçlı bilgi modeline dayanmaktadır [9]. Ayrıca Dünya üzerinde 3B nesnelerin semantik ve geometrik ilişkileri için en uygun çerçeveyi sağlamaktadır [18].

CityGML, aynı zamanda 3B kentsel modellerin görünüşlerinin temsili yanında kentlerde ve kentsel modellerde yer alabilecek amaca uygun topografik nesnelere yönelik sınıfları ve ilişkileri tanımlar. Kenti sadece bina yapılarıyla değil, bunun yanında yükseklik, bitki örtüsü, su kütlesi ve diğer fiziksel nesneleriyle birlikte tanımlar [26].

3.1.1 Tarihsel gelişimi

CityGML, SIG 3D üyeleri tarafından 2002 yılından bu yana geliştirilmektedir. Bu grup 2010 yılından itibaren GDI-DE (Almanya mekansal veri altyapısı) inisiyatifinin bir parçasıdır. 2010'dan önce SIG 3D, GDI NRW (Kuzey Ren-Vestfalya mekansal veri altyapısı) girişiminin üyesiydi [7], [9]. SIG 3D, birlikte çalışabilir 3B kent modelleri ve coğrafi görselleştirmenin geliştirilmesi ve ticari kullanımı üzerine çalışan; Almanya, Büyük Britanya, İsviçre ve Avusturya'dan 70'den fazla şirket, belediye ve araştırma enstitülerinden oluşan açık bir gruptur. SIG 3D çalışmalarının diğer sonucu da, aynı zamanda OGC içinde tartışılmakta olan Web 3D Servisi (3B sunum servisi) önerisidir [5], [9].

CityGML'in ilk başarılı uygulaması ve değerlendirmesi 2005'de GDI NRW'nin "Pilot 3D" projesinde gerçekleştirilmiştir. 2006 yılının başlarında EuroSDR tarafından gerçekleştirilen CityGML'e yönelik bir proje ile 3B kent modellerinin Avrupa çapında uyumlaştırılmasına başlanmıştır. 2006 yılında Haziran ayından Aralık ayına kadar, CityGML OpenGIS Web Hizmetleri 4'ün CAD/GIS/BIM iş parçacığı içinde kullanılmış ve değerlendirilmiştir. 2008 yılında CityGML 1.0.0 versiyonu, OGC standardı olarak benimsenmiştir [9].

Zaman içinde bu noktadan itibaren, CityGML Dünya çapında yayılmıştır. Almanya'nın birçok kenti ve diğer Avrupa ülkeleri (örn. Berlin, Köln, Dresden ve Münih vd.) kendi 3B modellerini oluşturmuşlardır. Fransa'da Bina3B (*Bati3D*) projesi, bir CityGML LoD2

profili tanımlar ve Paris ve Aix-en-Provence'in kent merkezleri olan Lille, Nantes ve Marsilya'ya ilişkin veriler sağlamaktadır. Aynı zamanda CityGML, Hollanda için 3B veri altyapısı ve 3B coğrafi bilgi standardı sağlamak için 3B pilot projelerde önemli rol oynamaktadır. Danimarka'daki kentlerin (LoD2 ve LoD3, kısmen LoD4) yanı sıra, Monako, Cenevre, Zürih, Leewarden gibi birçok Avrupa ülkesinde CityGML, LoD2 ve LoD3 veri temsili ve alış-verişi için kullanılır. CityGML, Avrupa mekansal veri altyapısının oluşturulması amaçlayan Avrupa Komisyonu'nun INSPIRE inisiyatifinin bina modelinden (INSPIRE version 2.0) önemli oranda etkilenmiştir. Asya'da Doha, Katar (LoD3) ve Yokohama'nın (LoD2) 3B kent modelleri, CityGML ile temsili ve veri alış-verişi sağlanmaktadır. Dahası, Malezya'da 3B mekansal veri altyapısı için CityGML önemli bir rol oynamaktadır [9].

3.1.2 CityGML 2.0'daki Ek Özellikler

CityGML 1.0 versiyonu, 3B kent modellerini temsil etmek ve değiştirmek için geniş çapta kullanılmaktadır. Fakat CityGML'in geliştirilmesi, özellikle SIG 3D tarafından hız kazandırılmış, devam etmekte olan bir süreçtir. 2.0 versiyonunda aşağıdaki eklentiler sağlanarak, Mart 2012'de uluslararası bir standart olarak OGC tarafından benimsenmiştir [9]:

- Köprü ve tüneller için tematik modeller, uygulamaya konmuştur. 1.0 versiyonunda, bu nesneler, genel nesneler olarak temsil edilmek zorundaydı. Birlikte çalışabilirlik ve semantik bakış açısından bu yaklaşım, farklı uygulamalarda aynı kavramlar muhtemelen farklı adlarla ifade edileceğinden yetersiz kalmaktaydı.
- Binalar için LoD0 temsili, 2B-3B entegrasyonunu desteklemek için eklenmiştir.
- Bütün kent nesneleri için iki yeni öznitelik, Araziye göre (*relativeToTerrain*) ve Suyu göre (*relativeToWater*) eklenmiştir [7].
- Dış Tavan Yüzeyi (*OuterCeilingSurface*) ve Dış Zemin Yüzeyi (*OuterFloorSurface*), binalar, köprüler ve tünellerin dış sınırlarının tematik yüzeyleri listesine eklenmiştir.
- Bir özniteliğin alabileceği tüm değerlere ilişkin listenin (kod listesi) referanslanmasına olanak veren bir mekanizma sağlanmaktadır. Bina işlevi

(*BuildingFunction*) ve çatı türü (*RoofType*) gibi öznitelik değerlerini numaralandıran kod listeleri, halihazırda 1.0 versiyonunda yer almaktadır. Bununla birlikte, özelleşmiş kod listeleri de oluşturulabilir. 2.0 versiyonunda bu listeler, her bir öznitelik için doğrudan referanslandırılabilir.

- Genel öznitelikler için grup mekanizması eklenmiştir. Öznitelik grubu bir isim ve bir kod boşluğu tarafından değerlendirilebilir. Dahası, ölçü birimi (örn. metre veya kilometrekare) GML Ölçü Türü (*MeasureType*) kullanılarak genel öznitelikler için doğrudan temsil edilebilir.

Version 2.0 geriye doğru version 1.0 ile uyumludur: 1.0'da geçerli olan tüm veri setleri, 2.0'da da geçerlidir, sadece ad alanı ve şema referansları uyarlanmak zorundadır [7], [9].

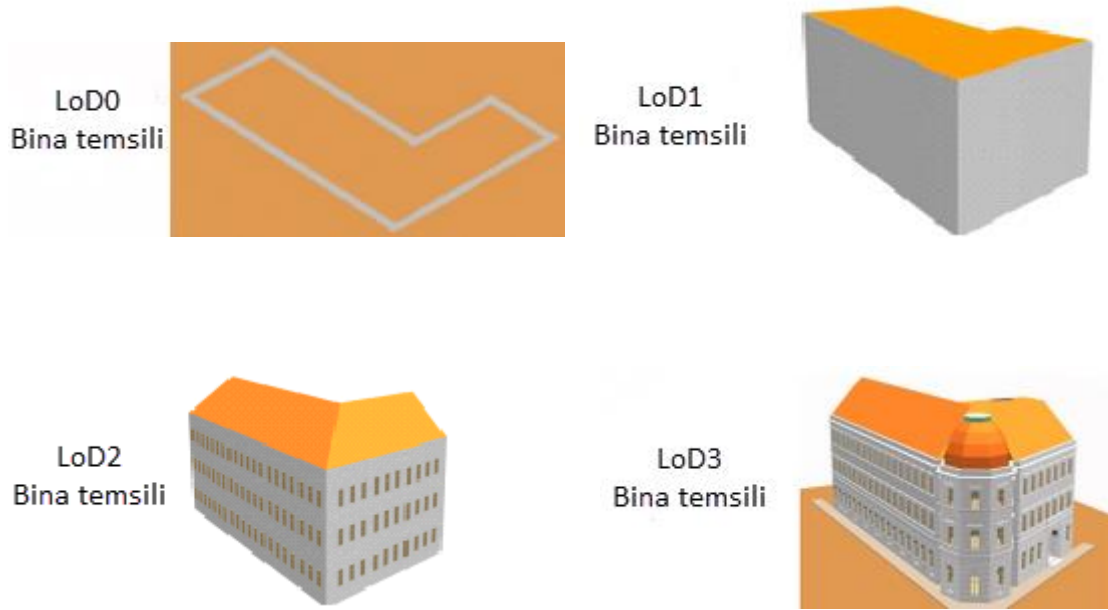
3. 2 CityGML'in Genel Özellikleri

3.2.1 Ayrıntı Düzeyleri (LoD)

CityGML, kent nesneleri için beş ayrıntı düzeyi (LoD0-LoD4) tanımlamaktadır [15], [36]. Genellikle beş ayrıntı düzeyinden biri, kent modeli için kullanılmaktadır. Fakat aynı nesneleri aynı anda farklı ayrıntı düzeylerinde temsil etmek mümkündür [36], [37]. Şekil 3.1'de aynı CityGML nesnesinin (burada bina) dört farklı ayrıntı düzeyinde görselleştirildiği bir örnek verilmektedir. Böylece, aynı nesnenin farklı çözünürlük derecelerine göre analizine ve görselleştirilmesine olanak sağlanmaktadır [8], [15]. Bu, tek bir ayrıntı düzeyi kapsamında kent nesnesinin geometrik ve semantik özelliklerinin tanımlanmasını gerektirir. Bu sayede, aynı kent nesnesi çoklu modeller ile tanımlanabilir [36].

Farklı ayrıntı düzeylerini yönetmedeki kritik nokta, veriler analiz edileceği ya da görselleştirileceği zaman, her nesne için - farklı ayrıntı düzeylerinde temsil edilse bile - yalnızca tek düzeyin dikkate alınmasını güvence altına almaktır. Birlikte güvenli olarak analiz edilebilen veya görselleştirilebilen nesne temsilleri topluluğu, görünüm (*view*) olarak adlandırılmaktadır. Bir görünümün farklı ayrıntı düzeyinden nesneler içerebildiği, fakat hiçbir zaman aynı nesneyi veya aynı alan parçasını (tekrarlı olarak) içermediği dikkate alınmalıdır. Aksi takdirde, görselleştirmede hata meydana gelir. Başka bir ifadeyle, aynı bina iki kez görüntülenir. Aynı nesnenin birden fazla dikkate

alınmasından dolayı analiz sonucu da hatalı olabilir. Örneğin, bir bölge içinde bina sayısının hesaplanması için her binanın sadece bir temsili dikkate alınmalıdır. Ayrıca, ayrıntı düzeyi yönetiminde tutarlı görünüm elde edilmesindeki başlıca zorluk, farklı ayrıntı düzeylerinde ki görünüm arasındaki ilişkilerin çok karmaşık olabilmesinden kaynaklanmaktadır [8].



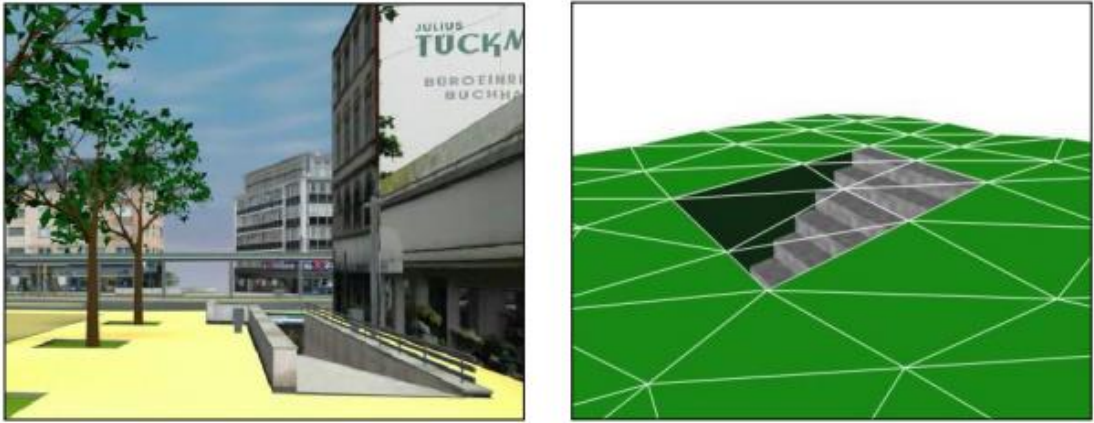
Şekil 3. 1 Bir binanın dört farklı ayrıntı düzeyinde temsili [8'den uyarlanmıştır]

3.2.2 Modülleştirme

CityGML veri modeli sanal 3B kent modelleri içinde en önemli nesne türleri için sınıf tanımlarını içermektedir. Bu sınıfların, bir çok uygulama alanı için gerekli ya da önemli olduğu saptanmıştır. Bununla birlikte, gerçekleştirmelerin standarda uymak amacıyla CityGML veri modelinin tamamını desteklemesi gerekmemektedir, fakat belirli bilgi gereksinimlerine göre standart yapıların alt kümesini kullanabilirler. Bu amaçla, CityGML veri modeli için modülleştirme uygulanmaktadır. CityGML on üç tane eklenti modülü sunmaktadır. Bu modüller: Görünüş (*Appearance*), Köprü (*Bridge*), Bina (*Building*), Kent Donatıları (*City Furniture*), Kent Nesnesi Grubu (*City Object Group*), Genel (*Generics*), Arazi Kullanımı (*LandUse*), Rölyef (*Relief*), Ulaşım (*Transportation*), Tünel (*Tunnel*), Bitki Örtüsü (*Vegetation*), Su Kütlesi (*WaterBody*) ve Dokulu yüzey (*TexturedSurface*) [9].

3.2.3 Kapatma Yüzeyi

Nesnelerin 2B temsillerinde kesişip 3B temsillerinde kesişmemesi, çoğu 3B modelleme uygulamasında yaygın problemlerden birisi olmuştur. Bu nesnelere örnek olarak, uçak hangarları, tüneller, yaya alt geçitleri ve yeraltından geçen demiryolları verilebilir. Gerçekte bu nesneler kapalı değildir. Fakat sel simülasyonları ve teknik altyapı veya çevre kirliliği kazaları sonucunda oluşan kirlenme alanları gibi farklı uygulama kapsamında nesnelerin hacimlerinin hesaplandığı durumlarda yüzeylerin kapatılması gerekmektedir [10], [33]. Bu problem, Şekil 3.2’de örnek olarak gösterilen CityGML’de kapatma yüzeyleriyle çözümlenmektedir. Bu kavram, hacim hesaplamaları yapmak ve sayısal arazi modelinde boşlukların oluşmasını önlemek amacıyla açık olan nesnelerin kapatılması için kullanılmaktadır. Buna rağmen, girişler gerçekliğe benzemesi için açık olarak işlem görmelidir [33].

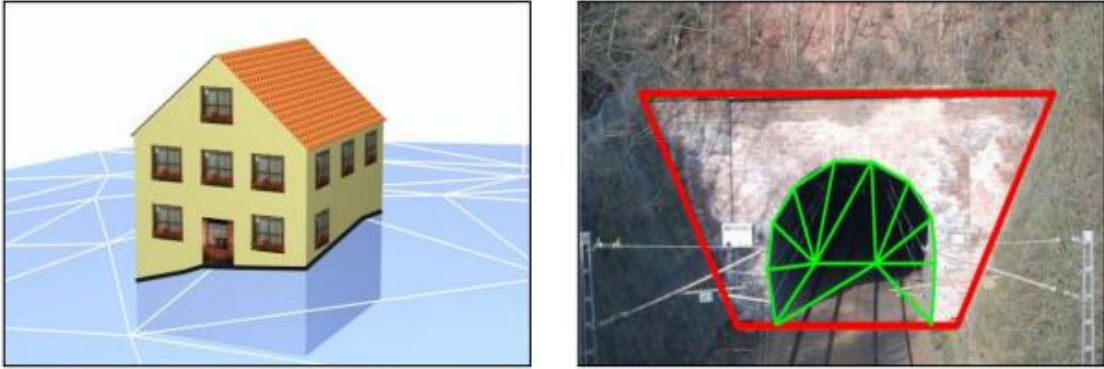


Şekil 3. 2 Yaya alt geçiti (solda) ve tünel (sağda) için kapatma yüzeyi temsili [9]

3.2.4 Arazi Kesişim Eğrisi (TIC)

Kent modellemede bina ve arazi entegrasyonu önemli hususlardan biridir [10]. Bu özellik, arazi modeli, farklı yükseklik değerine (Z) sahip bir 3B nesne ile bir araya getirildiğinde oluşan alışıldık teknik problemlerden birini açıklamaktadır. Prensip olarak 3B bir nesne, arazi modeline uymalıdır [38]. Ancak, sayısal arazi modeli ve 3B nesne, farklı ayrıntı düzeylerinde veya farklı veri sağlayıcıları tarafından oluşturulduysa, bunların kesişiminde önemli bir sorun ortaya çıkmaktadır [10], [38]. Bu problemin üstesinden gelmek için, binanın arazi arakesit eğrisi tanımlanmaktadır. Bu eğri, arazinin binaya temas ettiği yerdeki kesin konumu ifade eder ve binayı çevreleyen kapalı halka

ile gösterilir (Şekil 3.3) [10]. Ayrıca TIC kavramı, dokuların doğru konumlandırılmasını da sağlamaktadır.



Şekil 3.3 Bir binanın (solda) ve tünelin (sağda) arazi ile ara kesitleri [38]

3.2.5 Görünüşler

Görünüş, 3B kent modelleri için CityGML'de semantik ve geometrik modelin yanında üçüncü bileşen olarak kabul edilmektedir ve farklı fiziksel/görsel (örn. bina, ulaşım ve peyzaj) ve fiziksel olmayan (örn. kızıl ötesi radyasyon veya kirlilik alanları) nesnelerin nasıl gösterileceğini tanımlamaktadır. Görünüş, nesnenin görselleştirildiği yöntemi kontrol eder ve her ayrıntı düzeyi için ayrı olarak gerekli ayrıntıları ve dokuları belirlemektedir [33]. Bu konu, Bölüm 4. 3'te daha kapsamlı olarak incelenmiştir.

3.2.6 Uygulama Alanı Eklentileri (ADE)

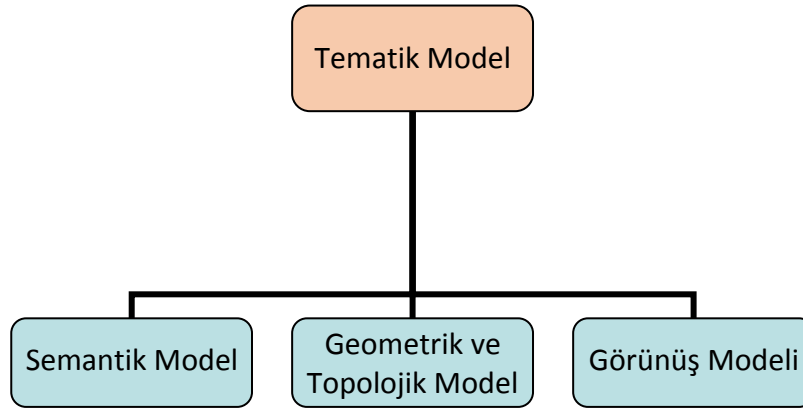
ADE, CityGML veri modeline olan eklentileri belirler. Bu tip eklentiler; mevcut CityGML sınıflarına yeni nesne türlerinin tanımı ya da binada oturan kişilerin sayısı gibi yeni özelliklerin eklenmesiyle oluşturulur. ADE'ler ile genel nesneler ve öznelilikler arasında fark, ADE'lerin kendi alan adı (*namespace*) ile tanımlı ek XML şema dosyası içinde tanımlı olmasıdır [29], [37], [39]. Bu dosya, doğrudan genişletilmiş CityGML modülünün XML Şeması tanımına dahil edilmelidir [5], [9], [37].

Bu yaklaşımın avantajı, eklentileri şeklen belirtmiş olmasıdır. Genişletilmiş CityGML örnek (*instance*) dokümanları, CityGML ve ilgili ADE şeması ile karşılaştırarak doğrulanabilir. ADE'ler belirli uygulama alanlarına ilgi duyan bilgi toplulukları tarafından tanımlanabilir (hatta standartlaştırılabilir). Birden fazla ADE, aktif olarak aynı veri setinde kullanılabilir [9], [37].

CityGML'DE MODELLEME ANLAYIŞI VE YAPISI

CityGML, daha önce ifade edildiği gibi 3B kent modellerinin yalnızca görünüşlerini temsil etmez aynı zamanda belirli semantik ve tematik özellikleri, taksonomi (aşamalı sınıflandırma) ve birleştirme hiyerarşilerini de ele almaktadır [8]. Dahası CityGML, modellerin geometrik, topolojik, semantik ve görünüş özellikleri ile ilgili olarak kent ve bölge modellerindeki amaca en uygun topografik nesneler için ilişkileri ve sınıfları tanımlar [5]. Çizelge 4.1'de Tematik model ile semantik, geometrik ve topolojik, görünüş model arasındaki ilişki oluşturulmuştur.

Çizelge 4. 1 CityGML'de modelleme yapısı



4.1 Tematik Model

Tematik model, sınıfların modülerleştirilmesine göre tanımlanmaktadır ve ana (*core*) modülden oluşur. Tüm eklenti modülleri, CityGML ana modülüne dayanmaktadır ve bu modüle bağımlıdır. Ana modül, CityGML veri modelinin temel kavramlarını ve bileşenlerini içermektedir. Belirli bilgi ihtiyaçlarına veya uygulama alanına göre, gerçekleştirmelerde ana modül ile bağlantılı olarak CityGML eklenti modüllerini

birleştirme tercih edilebilir. CityGML tematik modelinin her alanı, ayrı bir CityGML eklenti modülü ile kapsamaktadır. Böylece, eklenti modülleri CityGML'in geniş kapsamlı tematik veri modeli dikey bölünerek türetilebilir [9], [37].

Çizelge 4. 2 Tematik modüller ve tanımlı oldukları ayrıntı düzeylerinin içerdiği ilgili detay türleri [7].

Thematic module	Feature types	Feature types (cont'd.)
Core	_CityObject (LoD0-4)	
Relief	ReliefFeature (LoD0-4)	TINRelief (LoD0-4)
	MassPointRelief (LoD0-4)	BreaklineRelief (LoD0-4)
	RasterRelief (LoD0-4)	
Building	Building (LoD0-4)	BuildingPart (LoD0-4)
	BoundarySurface (LoD2-4)	BuildingFurniture (LoD4)
	Opening (Door, Window) (LoD3-4)	Room (LoD4)
	BuildingInstallation (LoD2-4)	IntBuildingInstallation (LoD4)
Tunnel	Tunnel (LoD1-4)	TunnelPart (LoD1-4)
	BoundarySurface (LoD2-4)	TunnelFurniture (LoD4)
	Opening (door, window) (LoD3-4)	HollowSpace (LoD4)
	TunnelInstallation (LoD2-4)	IntTunnelInstallation (LoD4)
Bridge	Bridge (LoD1-4)	BridgePart (LoD1-4)
	BoundarySurface (LoD2-4)	BridgeFurniture (LoD4)
	Opening (door, window) (LoD3-4)	BridgeRoom (LoD4)
	BridgeInstallation (LoD2-4)	BridgeConstructionElement (LoD1-4)
	IntBridgeInstallation (LoD4)	
Transportation	Road (LoD0-4)	Railway (LoD0-4)
	Track (LoD0-4)	Square (LoD0-4)
	TrafficArea (LoD2-4)	AuxiliaryTrafficArea (LoD2-4)
Water Body	WaterBody (LoD0-4)	WaterClosureSurface (LoD2-4)
	WaterGroundSurface (LoD2-4)	WaterSurface (LoD2-4)
Vegetation	SolitaryVegetationObject (LoD0-4)	PlantCover (LoD0-4)
CityFurniture	CityFurniture (LoD1-4)	
LandUse	LandUse (LoD1-4)	
Group	CityObjectGroup (LoD0-4)	
Generics	GenericCityObject (LoD0-4)	_GenericAttribute (LoD0-4)

Tematik modüller ve tematik modüllerin detay sınıfları Çizelge 4.2'de listelenmiştir. Ana modül, CityGML'de bütün detaylar için genel özniteliklere (yapım yılı, yıkım yılı vb.) sahip ve CityGML'de tüm detaylar tarafından devralınan temel sınıfları tanımlar. Rölyef (*Relief*) modülü, farklı ayrıntı düzeylerinde arazi temsiliinin farklı türlerini (TIN, 3B

noktalar, su toplama çizgileri ve raster) sağlamaktadır. Bina modülü, Bölüm 5’de daha kapsamlı olarak ele alınmıştır. Ulaşım (*Transportation*) modülü, tamamlayıcı elemanlar olarak kaldırımlar, orta şeritler, yeşil alan gibi Yardımcı Trafik Alanları (*Auxiliary Traffic Areas*) da dahil olmak üzere yol, demiryolu ve benzer detayları sağlar [7].

CityGML’in tematik modeli, SAM, alanlar (örneğin, binalar, köprüler ve tüneller), bitki örtüsü (tek nesneler ve aynı zamanda alansal ve hacimsel biyotoplar), arazi kullanımı, su kütleleri, ulaşım tesisleri ve kent donatıları gibi farklı tematik alanlar için geometri modeli kullanır. Henüz açıkça modellenmeyen başka nesneler, genel nesneler ve öznitelikler kavramı kullanılarak temsil edilebilir [9].

4.2 Semantik Model

CityGML için en önemli tasarım ilkelerinden biri, semantiklerin ve geometrik/topolojik özelliklerin tutarlı modellenmesidir. Semantik düzeyde gerçek dünya varlıkları; binalar, duvarlar, pencereler veya odalar gibi detaylar ile temsil edilmektedir. Bu tanım detaylar arasındaki öznitelikleri, ilişkileri ve birleştirme hiyerarşilerini (parça-bütün ilişkileri) de içermektedir. Dolayısıyla detaylar arasındaki parçası olma (*part-of*) ilişkisi, geometriyi dikkate almaksızın sadece semantik düzeyde elde edilebilmektedir. Bununla birlikte, mekansal düzeyde, geometri nesnelerine onların mekansal konumunu ve kapsamını temsil eden detaylar atanmaktadır. Böylece model, eşlenik nesnelerin ilişkiler ile bağlandığı semantik ve geometrik hiyerarşilerden oluşur. Bu yaklaşımın avantajı, tematik ve/veya geometrik sorgulamaları cevaplamak veya analizleri gerçekleştirmek için her iki hiyerarşi içinde ve arasında keyfi olarak dolaşılabilmesidir. Belirli bir nesne için her iki hiyerarşinin var olması durumunda, tutarlı olmaları gerekir. Başka bir ifadeyle, birbirleriyle eşleşmeli ve uyum sağlamalıdır. Örneğin, semantik düzeyde bir binanın duvarının iki pencere ve bir kapıya sahip olması durumunda, duvarı gösteren geometrinin aynı zamanda hem pencerelerin hem de kapının geometri parçalarını içermesi zorunludur [9].

CityGML’in semantik modeli, binalar, SAM, su kütleleri, ulaşım, bitki örtüsü ve kent donatıları dahil olmak üzere sanal 3B kent modellerindeki en önemli tematik nesne türleri için sınıf tanımlamalarını içermektedir [1], [5].

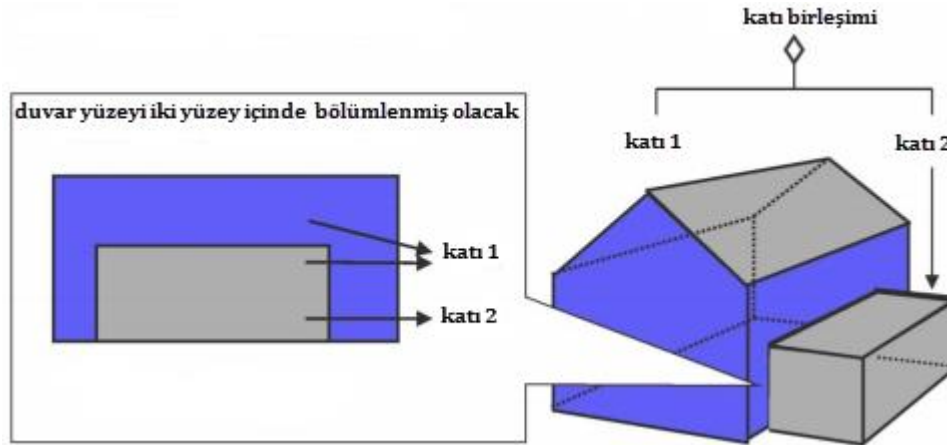
4.3 Geometrik ve Topolojik Model

Geometri modeli, 3B kent modellerinde mekansal nesnelerin geometrik ve topolojik özelliklerinin tutarlı ve homojen tanımlanmasına imkan sağlamaktadır [8]. Geometrik detay temsili, ISO 19107 Mekansal Şema (*Spatial Schema*)'nın bir gerçekleştirimi olan GML 3.1.1 tarafından sağlanan geometrik modeli esas almaktadır. Detaylar iyi bilinen Sınır Temsili (*Boundary Representation*) ile geometrik olarak temsil edilir [1], [5], [9], [10]. Binalar gibi hacimli detaylar, sınırlayıcı yüzeyleriyle tanımlı katı olarak temsil edilmektedir [7].

GML3'ün geometri modeli; kompleksler, karma geometriler veya birleşimleri (*aggregate*) oluşturmak için kombine edilebilen ilkelerden meydana gelmektedir. Her boyut için bir geometrik ilkel vardır; sıfır boyutlu bir nesne nokta (*Point*), bir boyutlu eğri (*Curve*), iki boyutlu yüzey (*Surface*) ve üç boyutlu katı (*Solid*)'dır. Her geometri, kendi koordinat sistemine sahip olabilir. Bir katı, yüzeyler ve yüzey eğrileri; bir yüzey, kenarlar ve bir kenar, noktalar tarafından sınırlandırılmaktadır [9], [10]. CityGML'de eğri, düz çizgi ile sınırlandırılmıştır, dolayısıyla sadece GML3 *LineString* sınıfını kullanmaktadır [9]. Ayrıca, CityGML'de yüzeyler, düzlemsel geometriyi tanımlayan çokgenler tarafından temsil edilmektedir, yani sınır ve tüm iç noktaların bütün koordinatlarının aynı düzlemde konumlanması gerekir [9], [10]. Benzer şekilde, sadece düz eğrilere izin verilir. Bir caminin çatısı gibi doğrusal olmayan yapılara düzlemsel yüzeyler ile yakınsanabilir. Geometrik temsile ait her konum için mutlak 3B koordinatlar açıkça verilmelidir. Bu durum, önceden GML tarafından öngörülmüştür [5]. Kenarlar, yüzeyler ve katılar sırasıyla, eğri geometrileri (*CurveGeometries*), yüzey geometrileri (*SurfaceGeometries*) ve katı geometrileri (*SolidGeometries*) için toplanabilir. Bu geometriler, uygulama nesnelerinin mekansal özelliklerini tanımlamak için kullanılmaktadır [10].

CityGML, topolojinin açıkça modellenmesini sağlamaktadır. Örnek olarak, detaylar ve diğer geometriler arasında geometri nesnelerinin paylaşımı verilebilir. Alanın bir parçası, yalnızca bir kez geometri nesneleri tarafından temsil edilmektedir ve tüm detaylar ya da bu geometri nesnesi tarafından tanımlanan ya da sınırlanan daha kompleks geometriler tarafından referanslanır. Böylece yinelemeden kaçınılmakta ve parçalar arasındaki topolojik ilişkiler sürdürülmektedir [1], [9]. Temel olarak üç durum

vardır. İlk olarak, iki detay mekansal olarak benzer geometriler tarafından tanımlanabilir. Örneğin, bir patika, hem ulaşım detayı hem de bitki örtüsü detayı ise patikayı tanımlayan yüzey geometrisi, hem ulaşım nesnesi hem de bitki örtüsü nesnesince referanslanır. İkincisi geometri, bir detay ve bir başka geometri arasında paylaşılabilir. Bir binanın duvarı olarak tanımlanan bir geometri, binanın geometrisini tanımlayan katı geometri ve duvar detayı ile iki kez referanslandırılabilir. Üçüncü olarak iki geometri, her ikisinin sınırındaki aynı geometriyle referanslandırılabilir. Örneğin, bir bina ve bitişindeki garaj, iki adet katı tarafından temsil edilebilir. Her iki katının temas ettiği yüzeyi tanımlayan alan, yalnızca bir kez temsil edilebilir ve her iki katı tarafından referanslanabilir. Şekil 4.1'de görülebileceği gibi bu durum, ilgili yüzeylerin bölümlenmesini gerektirir. Genel olarak, sınır temsili (*boundary representation*), yalnızca görsel yüzeyleri dikkate alır. Bununla birlikte, topolojik komşuluğu açıkça ifade edilir hale getirmek ve geri kalan birleşimde boşluk bırakmadan oluşturulan nesnenin bir kısmının silinmesi olanağı sağlamak için temas eden elemanlar kapsama dahil edilmiştir. Temasa izin verildiğinde, aynı alanın çoklu temsilinden kaçınmak amacıyla nesnelerin birbirine geçmesine izin verilmez. Bununla birlikte, CityGML'de topoloji kullanımı isteğe bağlıdır [9].



Şekil 4. 1 Bitişikteki garaj olan bir bina. Binanın ve garajın mekansal kapsamalarını temsil eden katılar arasındaki topolojik komşuluk, ortak duvar yüzeyini paylaşarak açık hale getirilebilir [5]

Birçok uygulama için nesne geometrisinin topolojik doğruluğu önemlidir. Örneğin, binanın hacmini hesaplamak için binayı sınırlandıran yüzeyler kapatılmalıdır. Ayrıca, Bina Parçaları (*BuildingParts*) gibi bitişik nesnelerin katıları temas etmek zorundadır, fakat onların iç kısımlarının birbirlerine geçmelerine izin verilmez. Çünkü boşluk,

yalnızca bir fiziksel nesne tarafından doldurulur. İç mekan navigasyonu uygulama alanında, odalar topolojik olarak bağlanmalıdır. Böylece rota planlama için 3B geometri ağına dönüştürülebilen bağlantı çizgesinin hesabına olanaklı hale gelir [5].

4.4 Görünüş Modeli

Mekansal özelliklere ek olarak, CityGML detaylarının, yüzeylerine ilişkin gözlemlenebilir özelliklerini ifade eden görünüşleri vardır. Görünüşler (bir cephenin görsel niteliğini gösteren dokular), görsel veriler ile sınırlı değil aksine kızılötesi radyasyon, gürültü kirliliği veya depremin tetiklediği yapısal gerilim gibi tema olarak adlandırılan isteğe bağlı kategorileri temsil eder [5], [7], [9], [12]. Bundan dolayı görünüşler, görselleştirmelerin yanı sıra analizleri de destekleyebilir [7]. Her ayrıntı düzeyi, belirli bir temalara yönelik bir görünüşe sahip olabilir. Görünüş, her bir yüzey geometrisi nesnesine yönelik verilerden oluşturulur. Tek bir yüzey nesnesi geometrisi, çoklu temalar için yüzey verilerine sahip olabilir. Benzer biçimde yüzey verisi, çoklu yüzey geometrisi nesneleri tarafından paylaşılabilir. Son olarak, yüzey veri değerleri, bir yüzey üzerinde sabit olabilir ya da yüzeydeki kesin konuma bağlıdır [9].

Bir görünüşü tanımlama ve yüzeyler üzerine eşleştirme mekanizması, bilgisayar grafik standartlarından (özellikle X3D) uyarlanmaktadır. Görünüşlerin iki türü desteklenmektedir: dönüşümler ile duvar veya çatı yüzeyler üzerine eşleştirilmiş parametrelerle ifade edilen dokular (*parameterized textures*) ve arazi yüzeyi üzerine izdüşürülen coğrafi referanslı dokular (*georeferenced textures*). İlk türe örnek, binanın foto gerçekçi görselleştirmesini elde etmek için tanımlı cephe ve çatı yüzeylerinin görüntüleridir. Arazi üzerine kaplanan bir ortofoto, dokunun ikinci türünü örneklemektedir [5]. Sonuç olarak, görünüşler ile sağlanan veri, sanal 3B kent modellerinde hem sunum hem de analiz için girdi olarak kullanılabilir. CityGML, her kent modeline keyfi sayıda tema için detay görünüşlerini destekler. CityGML'in görünüş modeli, kendi eklenti modülü *Görünüş (Appearance)* içinde paketlenmektedir.

Materyali temsil etmek için kullanılan görüntünün, bilgisayar grafikleri veya fotogrametriden doku eşleştirme yöntemleri kullanılarak duvar, çatı veya arazi yüzeyiyle düzgünce eşleştirilmiş olduğu varsayılmaktadır [12].

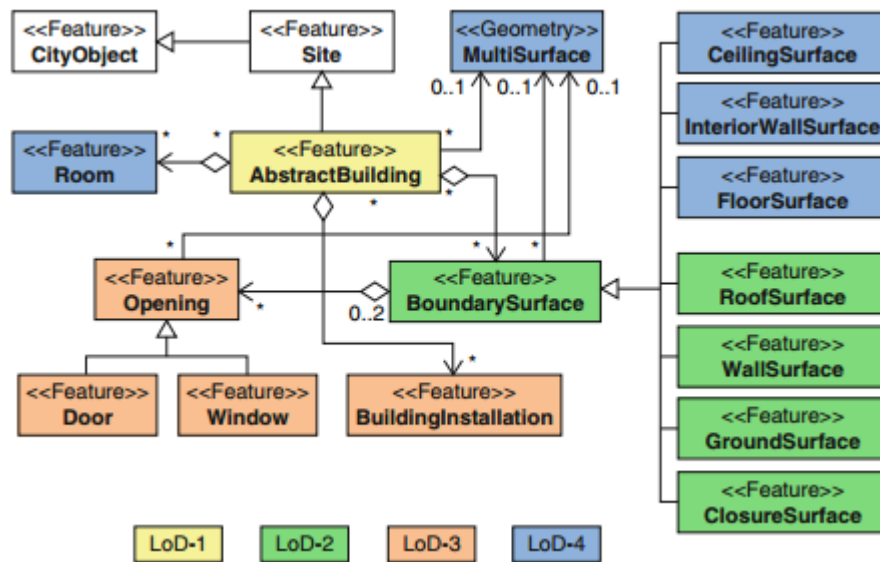
CityGML'DE BİNA MODELİ

3B kent modellerinin belirgin unsurları binalardır. Bina modeli, CityGML'in en kapsamlı tematik kavramlarından birisidir. Beş farklı ayrıntı düzeyinde (LoD0-LoD4) binalar (*Buildings*) ve bina parçalarının (*BuildingParts*) semantik, geometrik, topolojik ve görünüş özelliklerinin temsili mümkündür. CityGML'in bina modeli, *Building* tematik eklenti modülü tarafından tanımlanmaktadır [7], [9], [12].

Aynı zamanda bina modeli, CityGML'in en önemli bileşenidir. Binalar ve bina parçalarının geometrik (bina parçaları, odalar, duvar yüzeyleri, çatı yüzeyleri, kapılar, pencereler, vb.) ve semantik (detay türleri ve özellikleri) açıdan temsiline olanak sağlamaktadır [7], [12], [40]. Bu binalar ve parçaları, kent nesnesi (*CityObject*) üst sınıfından devralınan (kalıtım yoluyla aktarılan) ortak özniteliklere sahiptir. Binalar için sağlanan öznitelikler; sınıf, işlev, kullanım, yapım yılı, yıkım yılı, ölçülen yükseklik, çatı tipi ve yer üstünde ve altındaki katların sayısı ve yükseklikleridir. Bundan başka, OASIS tarafından yayınlanan XNAL/AL adres standartlarına göre temsil edilen bir ya da daha fazla adres binalara atanabilir [11]. Bazı semantik özellikler ve çoklu çözünürlükler, binalar için gösterilse de kavramların çoğu diğer modüllerdeki detaylar için de uygulanmaktadır. Binaları ve CityGML'deki diğer detayları temsil etmek için kullanılan geometrik model, 3B katılar ile binalar gibi hacimsel nesneleri temsil eden sınır temsili (B-Rep) modelleme yöntemine dayanmaktadır. Katının boyutları, katıya ait çıkışma ve boşluklar olmaksızın sınırlayıcı yüzeyler tarafından tanımlanır. CityGML'de tüm katılar, düzlemsel çokgenler ile sınırlandırılmıştır [12].

CityGML, anlaması bazen zor olan UML diyagramları ile biçimsel olarak ifade edilmektedir. Bu nedenle, somut örnekler aracılığıyla UML örnek diyagramları kullanılarak yapı ve kavramlar açıklanmaktadır [7].

UML standart notasyonları, CityGML bina modelini geliştirmek için kullanılmaktadır. CityGML, farklı uygulamalarda gerekli ayrıntı düzeylerine ek olarak kent modeli nesnelerini temsil etmek için kullanılan beş ayrıntı düzeyinde geliştirilmiştir. Ayrıntı düzeyleri (LoD), LoD0'dan LoD4'e numaralandırılır ve her ayrıntı düzeyinin farklı doğruluğu ve minimum mekansal nesne büyüklüğü vardır. Tek binalar, LoD1'de ortaya çıkmaya başlar. Bu nedenle, LoD1'den LoD4'e CityGML bina modelini sunmaktadır. Şekil 5.1 farklı renklerle bina ayrıntı düzeylerini temsil etmektedir [41].



Şekil 5. 1 CityGML’de bina modeli [41]

Bina modelinin UML diyagramı Ek-B’de gösterilmiştir. Modelin merkezi sınıfı, tematik *Site* sınıfı ve geçişli olarak en üstteki *CityObject* sınıfının alt sınıfı olan *AbstractBuilding*’dir. *AbstractBuilding* ya *Building* ya da *BuildingPart* ile özelleştirilmektedir. *BuildingParts*’tan oluşan *AbstractBuilding*, yine *AbstractBuilding* olduğu için keyfi derinlikte bir birleştirme hiyerarşisi gerçekleştirilebilir. *AbstractBuilding* sınıfı, en üstteki *CityObject* sınıfının alt sınıfı olarak, GML3 standart detay özniteliklerini (örn. *gml:name*) ve *ExternalReferences* gibi CityGML’e özgü öznitelikleri devralır. Dahası *AbstractBuilding* tarafından doğrudan kapsanmayan özellikler, *CityGML Generics* modülü tarafından sağlanan genel öznitelikler olarak veya CityGML uygulama alanı eklentisi (ADE) mekanizması kullanılarak modellenilebilir [9].

Fabrika sahası veya hastane kompleksleri gibi birçok bağımsız binadan oluşan bina kompleksleri, *City Object Groups* kavramı kullanılarak birleştirilmelidir. Kompleksin esas binası, karşılık gelen grup üyesinin rol adı, ana bina (*main building*) kullanılarak ifade edilebilir.

Hem *Building* hem de *BuildingPart* sınıfları *AbstractBuilding*'in özniteliklerini devralır. Bu parametreler kümesi, binanın üç boyutlu şeklinin kabaca oluşturulması için uygundur ve kadastro sistemi tarafından sağlanabilir. Ayrıca adres detayları, Bina (*Building*) veya Bina Parçaları (*BuildingParts*) için atanabilir [9].

AbstractBuilding sınıfının geometrik temsili ve semantik yapısı da Ek-B'de gösterilmektedir. Model, LoD0'dan LoD4'e ardışık olarak iyileştirilmektedir. Bu nedenle, bina modelinin tüm bileşenleri, her bir ayrıntı düzeyinde aynen temsil edilmez ve her ayrıntı düzeyinde tüm birleştirme düzeylerine izin verilmez. CityGML'de tüm nesne sınıfları, her ayrıntı düzeyi için önerilen minimum edinim ölçütlerine göre ayrıntı düzeyleriyle ilişkilendirilir. Bir nesne, farklı ayrıntı düzeylerinde farklı geometriler sağlanarak eş zamanlı olarak temsil edilebilir.

Çizelge 5. 1 *AbstractBuilding* sınıfının semantik temaları [9]

Geometrik / semantik tema	Özellik türü	LoD0	LoD1	LoD2	LoD3	LoD4
Bina taban alanı ve çatı kenarı	<i>gml:MultiSurfaceType</i>	•				
Bina iskeletinin hacimsel parçası	<i>gml:SolidType</i>		•	•	•	•
Bina iskeletinin yüzeysel parçası	<i>gml:MultiSurfaceType</i>		•	•	•	•
Arazi kesişim eğrisi	<i>gml:MultiCurveType</i>		•	•	•	•
Bina iskeletinin kıvrımlı parçası	<i>gml:MultiCurveType</i>			•	•	•
Bina parçaları	<i>BuildingPartType</i>		•	•	•	•
Sınır yüzeyleri	<i>AbstractBoundarySurfaceType</i>			•	•	•
Dıştaki bina tesisatları	<i>BuildingInstallationType</i>			•	•	•
Açıklıklar	<i>AbstractOpeningType</i>				•	•
Odalar	<i>RoomType</i>					•
İçteki bina tesisatları	<i>IntBuildingInstallationType</i>					•

Bina modelinin farklı geometrik ve semantik temalarının ayrıntı düzeylerindeki karşılığı Çizelge 5.1’de gösterilmektedir. LoD1-4 için bina hacmi, *gml:Solid* geometrisi ve/veya *gml:MultiSurface* geometrisi ile ifade edilebilir. Sayısal arazi modeliyle farklı kaynaklı binaları entegre etmekte kullanılan 3B arazi kesişim eğrisi tanımı da LoD1-4 için mümkündür [9].

Building sınıfı, *AbstractBuilding* sınıfının iki alt sınıfından biridir. Bir bina sadece bir (homojen) parçadan oluşuyor ise bu sınıf kullanılır. Örneğin, kat sayısı veya çatı tipi farklı olan yapısal bölütlerden (segment) oluşan bir bina, bir veya daha fazla ek bina parçasına sahip bir binaya ayrıştırılmalıdır. Şekil 5.2’de, bir ve iki parçadan oluşan bir bina örneğini göstermektedir. Binanın asıl parçasının geometrisi ve mekansal olmayan özellikleri, *Building* bileşke detayında temsil edilmelidir [9].



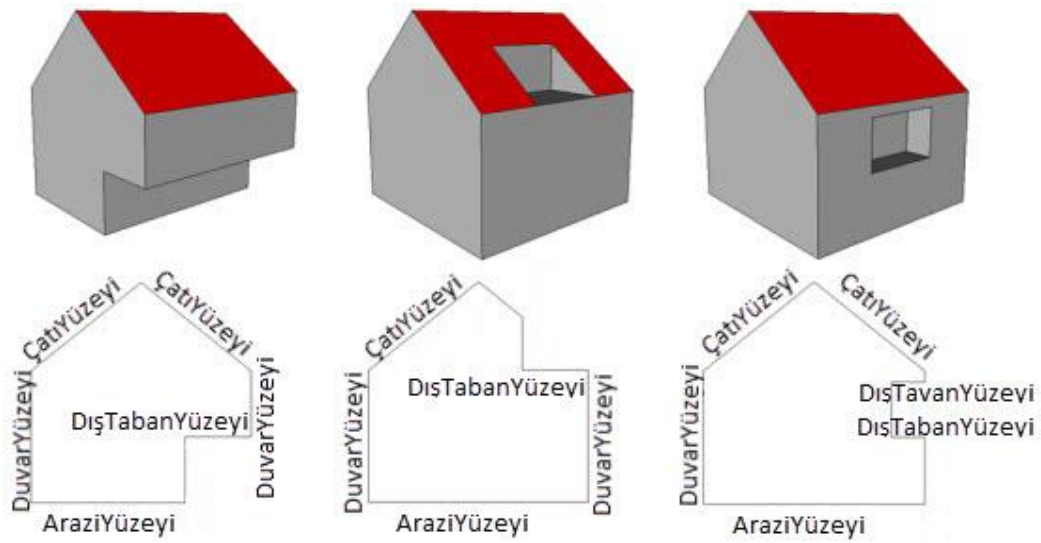
Şekil 5. 2 Bir ve iki bina parçası içeren bir bina örneği [12]

Bina tesisatı (*BuildingInstallation*), bir binanın sahip olduğu parça (*BuildingPart*) kadar önemli olmayan dış bileşenidir, fakat binanın dış özelliklerini fazlasıyla etkiler. Örneğin, bacalar, merdivenler, antenler, balkonlar veya yürüme yolları üzerine eklenen çatılar. Bir *BuildingInstallation* sınıfı isteğe bağlı olarak sınıf, işlev ve kullanım özniteliklerine sahiptir. Yalnızca bir kez oluşturulabilen sınıf özniteliği, tesisatın genel sınıflandırmasını ifade eder. İşlev ve kullanım öznitelikleriyle, bina tesisatının adlandırılan ve gerçek işlevleri tanımlanabilir. Bu üç öznitelik için uygun değerler listesi, bir kod listesinde belirtilebilir. Bir bina tesisatının geometrik temsili için GML alt kümesinden isteğe bağlı bir geometri nesnesi kullanılabilir. Alternatif olarak geometri, dolaylı geometri (*ImplicitGeometry*) nesnesi olarak verilebilir. Dolaylı geometri kavramının ardından bir

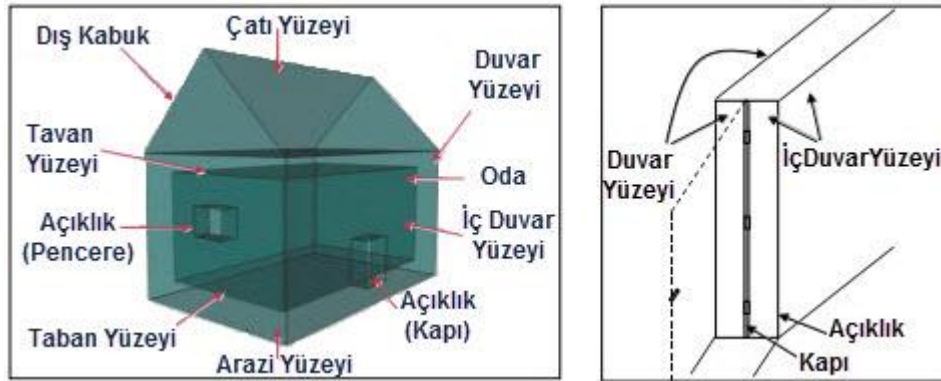
prototip bina tesisatının geometrisi, lokal koordinat sisteminde sadece bir kez depolanır ve diğer bina tesisatı detaylarınca referanslandırılır. Bina tesisatının görünür yüzeyleri, sınır yüzeyleri kavramı kullanılarak semantik olarak sınıflandırılabilir. Bir bina tesisatı nesnesi, tam olarak bir bina veya bina parçasının nesnesi ile tekil olarak ilişkide olmalıdır [9].

Sınır Yüzeyi (*BoundarySurface*), birkaç tematik sınıf için soyut üst sınıftır. Bir binanın dış iskeleti, odaların görünen yüzeyleri ve de hem dıştaki hem de içteki bina tesisatını yapılandırır. Sınır Yüzeyi (*BoundarySurface*) sınıfı, Kent Nesnesi (*CityObject*) sınıfının bir alt sınıfıdır ve bu nedenle GML3 standardı detay öznitelikleri (örn. *gml:name*) gibi bütün öznitelikleri devralır ve CityGML Dış Referanslar (*ExternalReferences*) gibi spesifik öznitelikleri belirler. Sınır Yüzeyi (*BoundarySurface*), Çatı Yüzeyi (*RoofSurface*), Duvar Yüzeyi (*WallSurface*), Arazi Yüzeyi (*GroundSurface*), Dış Tavan Yüzeyi (*OuterCeilingSurface*), Dış Taban Yüzeyi (*OuterFloorSurface*), Kapatma Yüzeyi (*ClosureSurface*), Taban Yüzeyi (*FloorSurface*), İç Duvar Yüzeyi (*InteriorWallSurface*) ve Tavan Yüzeyi (*CeilingSurface*) tematik sınıfları türetilir. Bina yüzeylerinin tematik sınıflandırması, Şekil 5.3 (dış bina iskeleti) ve Şekil 5.4 (ek iç yüzeyler)'te gösterilmektedir. LoD2 ve LoD4 arasında her bir ayrıntı düzeyinde, Sınır Yüzeyi (*BoundarySurface*) sınıfının geometrisi farklı, *gml:MultiSurface* geometrisi tarafından tanımlanabilir. LoD3 ve LoD4'de, Sınır Yüzeyi (*BoundarySurface*) kapılar ve pencereler gibi açıklıklar (*Openings*) içerebilir. Açıklıkların geometrik konumu, topolojik olarak *gml:MultiSurface* geometrisinin bir yüzey bileşeni (örn: *gml:Polygon*) içerisinde yer alıyorsa bu açıklık, yüzey içerisinde boşluk (*hole*) olarak gösterilmelidir. Boşluk, karşılık gelen yüzey geometrisi içerisinde iç halka olarak gösterilmektedir. GML3'e göre noktaların ters sırayla belirtilmiş olması gerekir (yüzey normal vektörünün aksi yönünde bakıldığında dış sınırlar saat ibresinin tersi ve iç sınırlar saat ibresi yönünde). Böyle açıklıklar, eğer Kapı (*Door*), Pencere (*Window*) veya Kapatma Yüzeyi (*ClosureSurface*) tarafından kapatılırsa, dış sınırları, çevreleyen yüzeylerin (boşluğunu ifade eden) iç halkası olarak aynı noktalardan oluşur. Açıklığın pervaz (*embrasure*) yüzeyleri, ilgili bitişik sınır yüzeyine (*BoundarySurface*) aittir. Örneğin, eğer bir kapı bu açıklığı kapatıyorsa, kapının bir kenarı üzerindeki pervaz yüzeyi, iç duvar yüzeyine ve diğer tarafta duvar yüzeyine aittir [9].

Açıklıklar (*Opening*) sınıfı, dışarıda kapılar veya pencereler ya da içeride duvarlar ve çatılar gibi semantik olarak tanımlanan açıklıklar için soyut bir üst sınıftır. Açıklıklar, sadece LoD3 veya LoD4'ün modellerinde bulunmaktadır. Her açıklık, çoklu yüzey (*gml:MultiSurface*) geometrisi ile ilişkilidir. Alternatif olarak geometri, dolaylı geometri (*ImplicitGeometry*) nesnesi olarak verilebilir. Dolaylı geometri kavramının ardından bir prototip açıklık, sadece bir kez lokal koordinat sisteminde saklanır ve diğer açıklık detaylarınca referanslandırılır [9].



Şekil 5.3 Bina dış iskeletinin sınır yüzeyi (*BoundarySurface*) sınıfı örneği [12]

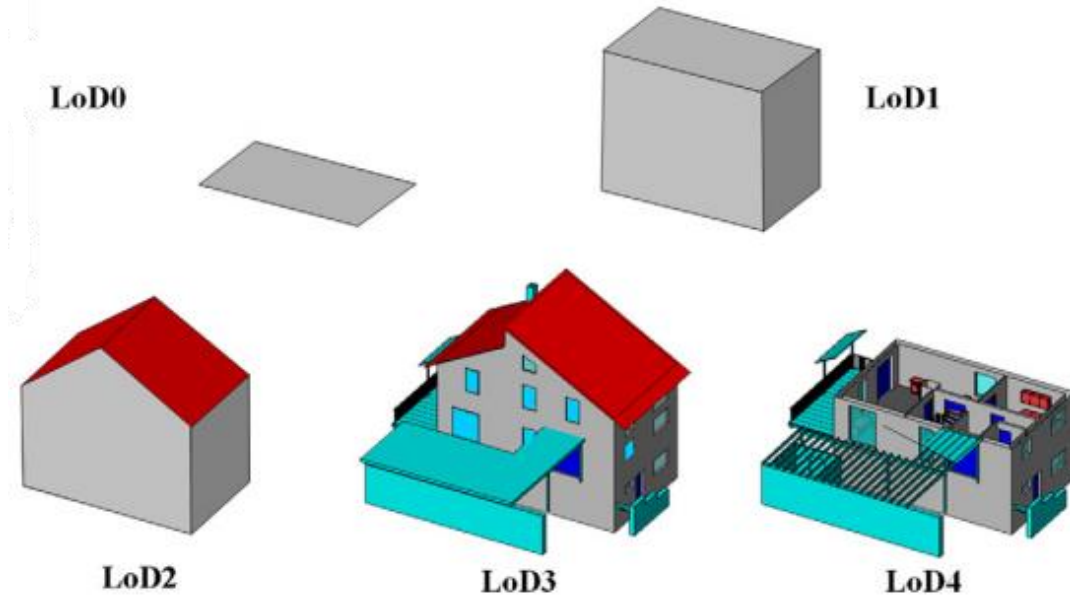


Şekil 5.4 Sınır yüzeylerinin (solda) özellikle açıklıklar (sağda) için sınıflandırılması [9]

5.1 Binaların Farklı Ayrıntı Düzeylerinde (LoD) Modellenmesi

CityGML'de binalar için beş farklı ayrıntı düzeyi tanımlanmaktadır (Şekil 5.5). En düşük düzey olan LoD0'da binalar 2.5B (2B geometri + yükseklik özniteliği) çokgenler ile temsil edilir. LoD1, çatı yapıları ve dokuları içermeyen bina blok modelidir. Buna karşılık LoD2'de binalar, farklılaştırılmış çatı yapıları ve dokulara sahiptir. LoD3, ayrıntılı duvar

ve çatı yapıları, balkonlar, duvar bölmeleri ve çıkıntılar ile mimari modelleri gösterir [37]. Yüksek çözünürlüklü dokular, bu yapılar üzerine giydirilebilir. LoD4, LoD3 modelin odalar, iç odalar, merdivenler ve mobilyalar gibi iç yapılar eklenerek oluşturulmaktadır [10].



Şekil 5. 5 CityGML tarafından sağlanan beş ayrıntı düzeyi (LoD) [9]

Ayrıntı düzeyleri, nesnelerin farklı doğrulukları ve minimum boyutlarına göre de nitelendirilmektedir (Çizelge 5.2). Bu standartta verilen doğruluk gereksinimleri, tartışılabilir. Doğruluk, mutlak 3B nokta koordinatlarının standart sapması (σ) ile tanımlanmaktadır. Bağıl 3B nokta doğruluğu, CityGML'in gelecek versiyonlarına eklenecektir ve genel anlamda mutlak doğruluktan çok daha yüksek olacaktır. LoD1'de noktaların konum ve yükseklik doğruluğu, en az 5 m olmalıdır, bununla birlikte minimum en az 6 m $\times$ 6 m taban alanına (*footprint*) sahip bütün nesneler dikkate alınmalıdır. LoD2'nin konum ve yükseklik doğruluğunun en az 2 m olması gerektiği önerilmiştir. Bu ayrıntı düzeyinde, minimum 4 m $\times$ 4 m taban alanına sahip bütün nesneler göz önünde bulundurulmalıdır. LoD3'de bu doğrulukların ikisi de 0.5 m olmalıdır ve minimum taban alanının 2 m $\times$ 2 m olabileceği önerilmektedir. Son olarak, LoD4'ün konum ve yükseklik doğruluğu 0.2 m veya daha az olmalıdır. Beş ayrıntı düzeyinde sınıflandırma, 3B kent modeli veri setlerinin kalitesini değerlendirmek için kullanılabilir. Ayrıca, ayrıntı düzeylerini sınıflandırma, veri setlerini karşılaştırılabilir hale getirmekte ve onların entegrasyonu için destek sağlamaktadır [9].

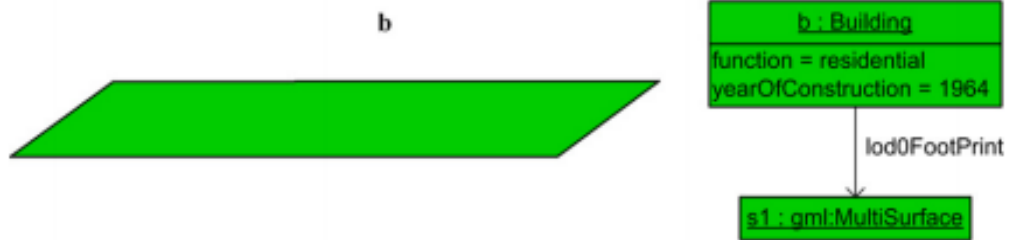
Çizelge 5. 2 CityGML’de ayrıntı düzeyleri için belirlenmiş çözünürlük ölçütleri [9]

	LoD0	LoD1	LoD2	LoD3	LoD4
Model ölçeği tanımlaması	bölgesel, peyzaj	kent, bölge	kent, kent idari birimleri, projeler	kent idari birimleri, mimari modeller (dış görünüş), kent simgesi	mimari modeller (iç görünüş), kent simgesi
Doğruluk sınıfı	en düşük	düşük	orta	yüksek	en yüksek
Mutlak 3B nokta doğruluğu	LoD1’den daha düşük	5 / 5m	2 / 2m	0.5 / 0.5m	0.2 / 0.2m
Genelleştirme	en fazla genelleştirme	genelleştirilmiş detaylar olarak nesne blokları > 6 × 6m / 3m	genelleştirilmiş detaylar olarak nesneler > 4 × 4m / 2m	gerçek detaylar olarak nesneler > 2 × 2m / 1m	yapı elemanları ve açıklıklar gösterilir.
Bina tesisatları	hayır	hayır	evet	temsili dış detaylar	gerçek nesne biçimi
Çatı yapısı/temsili	evet	düz	farklılaştırılmış çatı yapıları	gerçek nesnelerden	gerçek nesne biçimi
Çatı çıkıntıları	evet	hayır	evet (biliniyorsa)	evet	evet
Kent donatıları	hayır	önemli nesneler	prototip, genelleştirilmiş nesneler	gerçek nesne biçimi	gerçek nesne biçimi
Tek bitki örtüsü nesnesi	hayır	önemli nesneler	prototip, 6 m’den yüksek	prototip, 2 m’den yüksek	prototip, gerçek nesne biçimi
Bitki örtüsü	hayır	> 50 × 50m	> 5 × 5m	< LoD2	< LoD2

Binalar için LoD0 temsili, CityGML 2.0 versiyonunda uygulanmaya başlanmıştır. Bir bina eş zamanlı olarak birden fazla ayrıntı düzeyi ile temsil edilebilir. Geometri temsili, (0’dan 4’e) artan ayrıntı düzeyi ile daha ayrıntılı hale gelmektedir. Ancak, buradaki odak noktası, yapısal özellikler üzerinedir ve geometrik ve semantik yapı açısından tutarlı bir şekilde ayrıştırılan modeller incelenmiştir. Genel olarak, CityGML daha esnektir ve çok çeşitli temsiller sağlar. Bununla birlikte, belirli bir ayrıntı düzeyine hangi semantik nesnenin dahil edilmesi gerektiği kesin kurallar ile belirlenmediğinden CityGML’in esnekliği, bir eksiklik olarak görülebilir. Örneğin, yalnızca tek bir yapı detayına sahip, bütünüyle geometrik olarak ayrıntılı bir model, geçerli bir LoD3 modelidir [9].

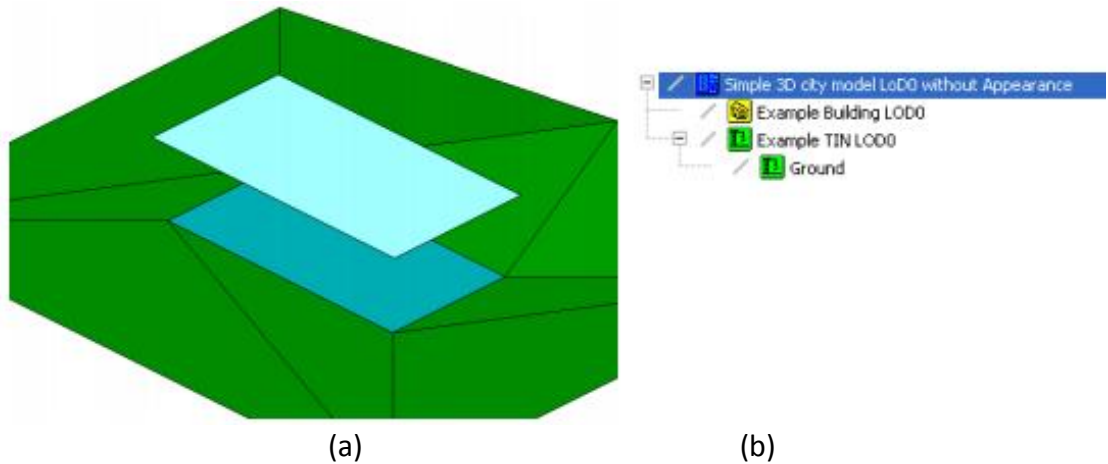
5.1.1 Ayrıntı Düzeyi 0 (LoD0)

LoD0'da bir bina, ya çatı seviyesi yüksekliğinde ya da zemin seviyesi yüksekliğinde 2.5B çokgenler ile temsil edilir. Semantik, ilgili öznitelik değerine sahip bir bina özel nesnesi (*buildinginstance*) ile modellenmektedir [7] (Şekil 5.6).



Şekil 5. 6 LoD0'da bina temsili: geometrik temsil (2.5B çokgen) ve UML örnek diyagramı [7]

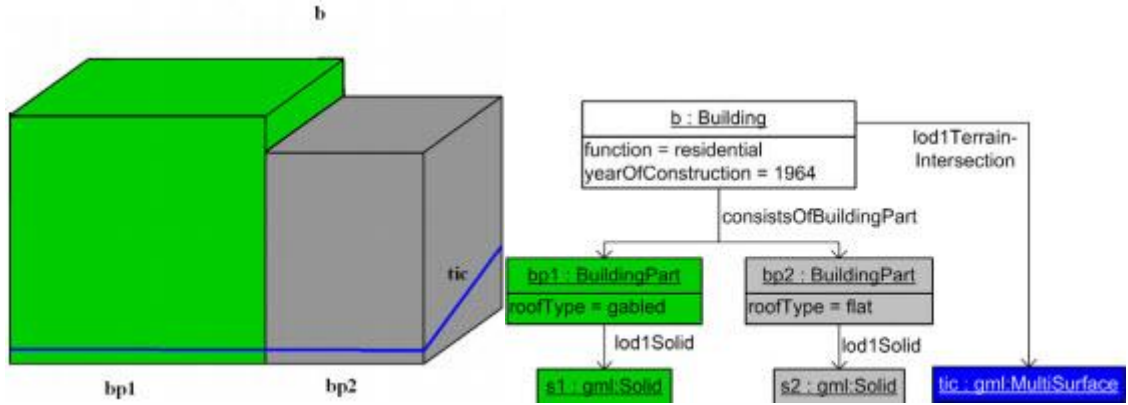
Örnek'te (Şekil 5.8) LoD0 Düzensiz Üçgen Ağı (*TIN*) (*dem:TINRelief*) ile temsil edilen bir arazi modelinin çevrelediği LoD0'daki bir bina gösterilmektedir. Hem taban alanı (footprint) hem de çatı kenarı temsili binaya atanmaktadır (*bldg:lod0FootPrint* ve *bldg:lod0RoofEdge*). Bunlar, yüzeylerine ait 3B koordinatların yükseklik değerlerinin aynı olmak zorunda olduğu (yatay yüzeyler) *gml:MultiSurface* geometrisi ile tanımlanmaktadır. Taban alanı (koyu mavi renkli), binaya ilişkin en düşük yükseklikte konumlanırken, çatı kenarı temsili (açık mavi renkli), çatı seviyesinde (saçak yüksekliği) ve dolayısıyla arazi üzerinde yer almaktadır. Binaya xAL adres şeması kullanılarak adres bilgileri atanmaktadır. Şekil 5.7'de LoD0 içerilen semantik varlıklarının hiyerarşisi betimlenirken, solda geometrik temsili gösterilmektedir [9].



Şekil 5. 7 LoD0'da CityGML bina modeli (taban alanı ve çatı kenarı) ve arazi modeli (a) 3B geometri ve (b) model hiyerarşisi [9].

5.1.2 Ayrıntı Düzeyi 1 (LoD1)

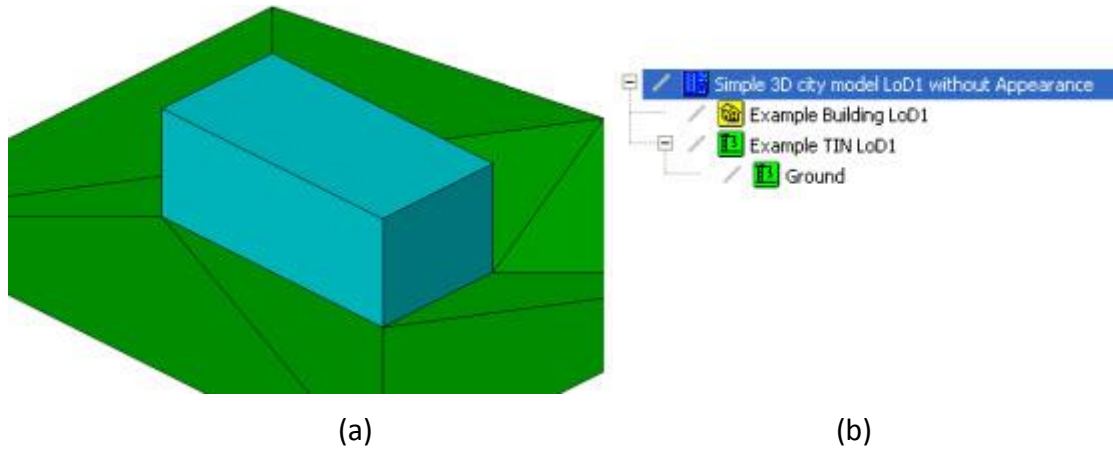
LoD1’de bina, katı model ya da çok yüzeyli bir blok model olarak temsil edilir. Şekil 5.8, LoD1 bina örnek (*instance*) diyagramı ve karşılık gelen bina yapısı gösterilmektedir. Bina (*Building*) farklı bina parçalarına (*BuildingParts*) ayrıştırılabilir. Bu bina bileşenlerinin geometrik (örn. farklı yükseklik) ve tematik özellikleri (örn. inşaat yılı) farklı olabilir. Binalar ve bina parçaları, potansiyel olarak aynı mekansal ve mekansal olmayan özelliklere sahiptir. Bir parçanın öznitelik değerleri, sadece o parça için geçerlidir oysa bir binanın öznitelik değerleri tüm parçalara aittir. Parçalar mevcut ise, bina geometrisi yalnızca parçalar ile temsil edilmelidir. Şekil 5.9’da görüldüğü gibi bir *b* binası, *bp1* ve *bp2* olmak üzere iki parçadan oluşmaktadır. Her parça, kendi çatı tipine sahiptir ve katı (*solid*) ile temsil edilmektedir. Çatı tipi, binanın LoD1’deki temsiline değil, gerçek şekline yöneliktir. Bu nedenle LoD1’deki temsili daima düz, yatay bir çatı olsa da, öznitelik değeri “üçgen çatı” (*gabled roof*) olabilir. Hem *bp1* hem de *bp2* parçasının temas ettiği yüzey, topolojik kavramlar kullanarak doğrudan temsil edilebilir. Her ayrıntı düzeyinde mevcut olan arazi kesişme eğrisi (TIC - arazi yüzeyinin binanın dış sınır yüzeyine temas ettiği kısımdaki çizgi) doğrudan temsil edilir. Böylece arazi ile binanın sorunsuz entegrasyonu sağlanır [7].



Şekil 5. 8 LoD1’de bina temsili: geometrik temsili (blok model) ve UML örnek diyagramı [7]

Şekil 5.9, LoD1’de hem bina hem de arazi modelini göstermektedir. Bina, blok model ile temsil edilmektedir. Bina geometrisi, dış iskeleti altı tane yüzey (*gml:Polygon*) ile sınırlanan bir katı (*gml:Solid*) olarak tanımlanmaktadır. Geometriye ek olarak bina, daha fazla tematik öznitelik ile zenginleştirilmektedir. Bu özniteliklerin bazıları

nümeriktir ve CityGML kod listesi mekanizmasının kullanımını gösterir. Kodlanmış öznitelik değerleri, SIG 3D Konsorsiyumu tarafından önerilen kod listelerinden alınır [9].

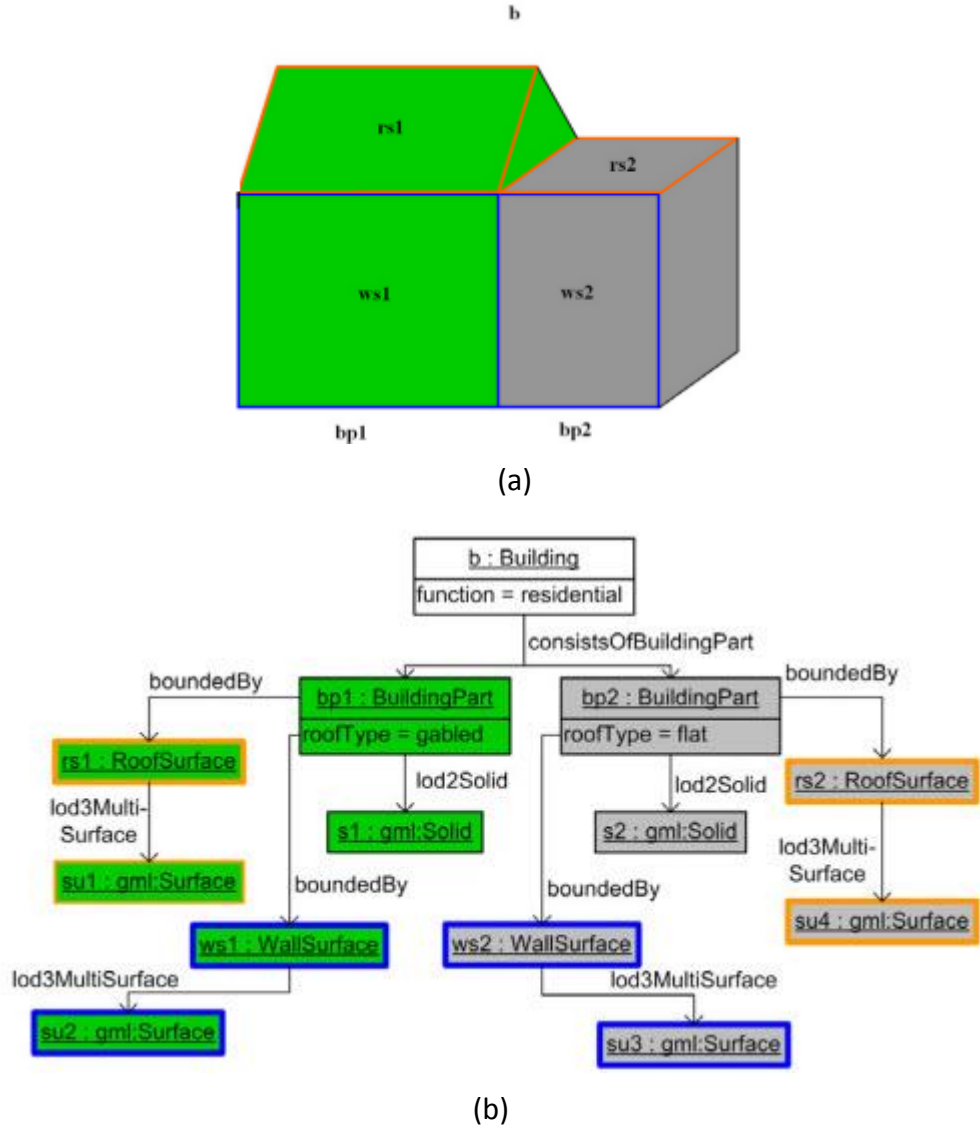


Şekil 5. 9 LoD1’de CityGML bina modeli: (a) 3B geometri ve (b) model hiyerarşisi [9].

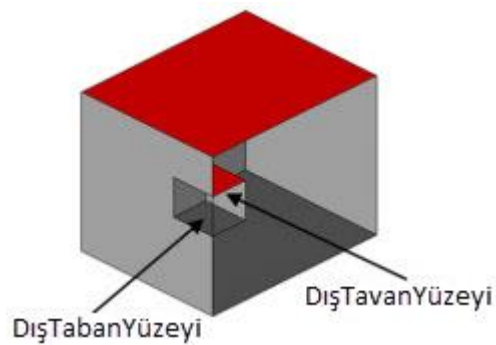
5.1.3 Ayrıntı Düzeyi 2 (LoD2)

LoD2, LoD1’e genelleştirilmiş çatı yapılarını ekler (Şekil 5.10). Buna ek olarak, bir binanın sınır yüzeylerinin temsilde, tematik detaylar kullanılabilir. Düşey duvar yüzeyleri, *WallSurface*; üst taraftan binayı örten yüzeyler, *RoofSurfaces* ve zeminden bina alt tarafını sınırlayan yatay yüzeyler, *GroundSurfaces* ile temsil edilir. Örneğin, Şekil 5.10’da UML örnek diyagramında üçgen çatının parçası olarak *rs1* çatı yüzeyi ve düz çatıyı tanımlayan *rs2* çatı yüzeyi, geometrik temsili yüzey olan detaylar ile tanımlanır. Aynı yaklaşım, *ws1* ve *ws2* duvar yüzeyleri için de geçerlidir [7].

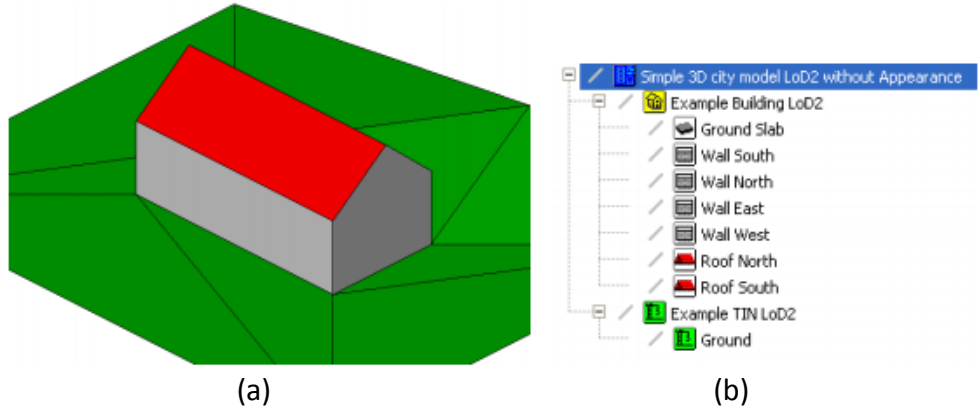
Uçak hangarları ve barınak gibi tamamen kapalı olmayan binalar için açık kenarları kapatmak ve geometrik olarak kapalı hacimsel nesne temsili sağlamak amacıyla kapatma yüzeyleri (*ClosureSurfaces*) kullanılmaktadır. Kapatma yüzeyleri, gerçek dünyada karşılığı olmayan nesnelerdir. Hacim hesaplamalarında dikkate alınırken görselleştirme gibi gerekli ya da uygun olmadıkları durumlarda ihmal edilirler. CityGML 2.0 versiyonunda binanın dış sınırına ilişkin tematik yüzeyler listesine, Dış Tavan Yüzeyi (*OuterCeilingSurface*) ve Dış Taban Yüzeyi (*OuterFloorSurface*) olmak üzere iki tane tematik yüzey eklenmiştir (Şekil 5.11). Buna ilişkin örnek olarak, tavan veya sundurma (avlu) zeminini tanımlayan yüzeyler verilebilir [7].



Şekil 5. 10 LoD2’de bina temsili: (a) geometrik temsil, (b) UML örnek diyagramı [7]



Şekil 5. 11 Bina modelinin sınırındaki iki ek tematik yüzey türü. Örnekte, Dış Tavan Yüzeyi yukarıdan, Dış Taban Yüzeyi ise aşağıdan sundurmayı sınırlar [7]



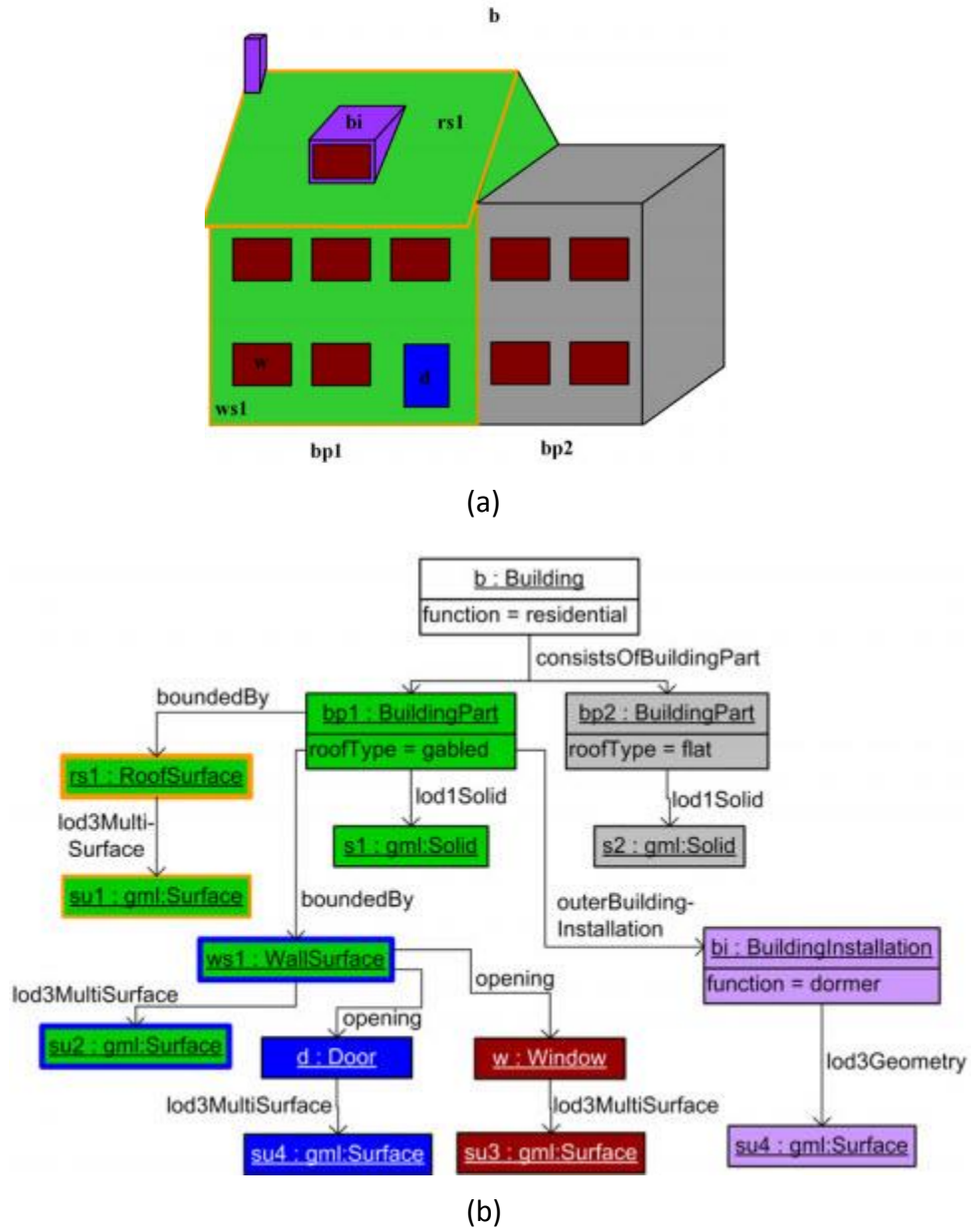
Şekil 5. 12 LoD2’de CityGML bina modeli: (a) 3B grafik ve (b) model hiyerarşisi [9]

LoD2 örneğinde (Şekil 5.12) model, gerçek çatı yapısını yansıtır ve dış bina iskeleti yüzeylerini (*bldg:RoofSurface*, *bldg:WallSurface* ve *bldg:GroundSurface*) semantik olarak sınıflandıran sınır yüzeylerini (*bldg:boundedBy*) içerir. Tematik sınır yüzeylerine ek olarak, bina geometrisi LoD2 katı geometrisi (*gml:Solid*) kullanılarak da tanımlanmaktadır [9].

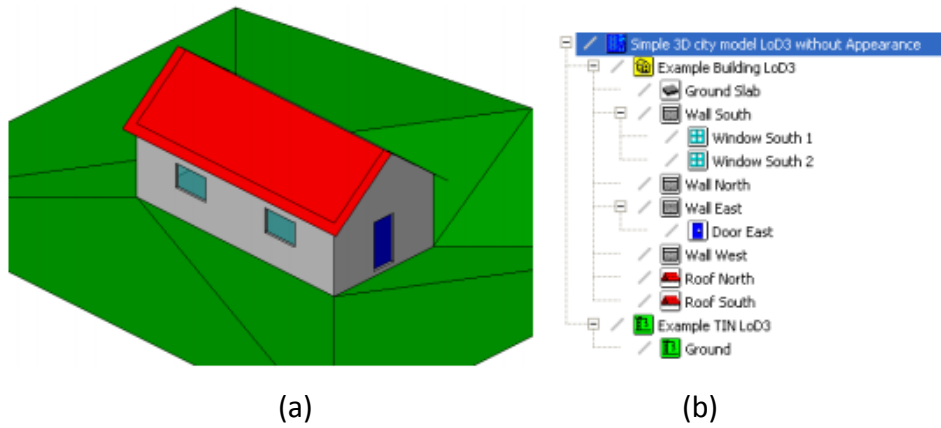
5.1.4 Ayrıntı Düzeyi 3 (LoD3)

LoD3, LoD2’nin açıklıklar (pencereler, kapılar), ayrıntılı çatı yapıları (çatı pencereleri, bacalar, çatı çıkıntıları) ve ayrıntılı cephe yapılarıyla genişletilmesiyle elde edilir. Bu nesneler, kendi öznitelikleri ve (yüzey) geometrisine sahip detaylar olarak temsil edilir. Şekil 5.13’te CityGML’de işlev (*function*) özniteliğinin değeri, çatı penceresi (*dormer*) olan ve bina tesisatı (*BuildingInstallation*) olarak temsil edilen pencere (*w*), kapı (*d*) ve çatı penceresi (*bi*)’ne sahip örnek bir LoD3 binası betimlenmektedir. Örnek diyagramında pencere ve kapılar, ilgili duvar/çatı yüzeylerine ve çatı penceresi, bina parçasına atanır. Her bir detay, geometrik özelliklerini temsil eden yüzey geometrisine sahiptir [7].

LoD3 bina modeli örneği, Şekil 5.14’te gösterilmektedir. Geçerli bir katı geometrisi elde etmek için, çatı yüzeyleri, geometrik olarak çatı döşemesi ve çatı çıkıntı parçaları halinde ikiye bölünür. Sadece çatı döşemesinin geometrisi katı olarak referans alınır. Birkaç yüzeyden (örn. pervazlar) oluşan duvarlar, bileşke yüzey (*gml:CompositeSurface*) olarak modellenir ve daha sonra bina katısı ile referanslandırılır. Kapı veya pencerelere yönelik açıklıkları içeren sınır yüzeyler, bir dış ve bir kaç iç doğrusal halkaya sahip çokgenler ile modellenmektedir [9].



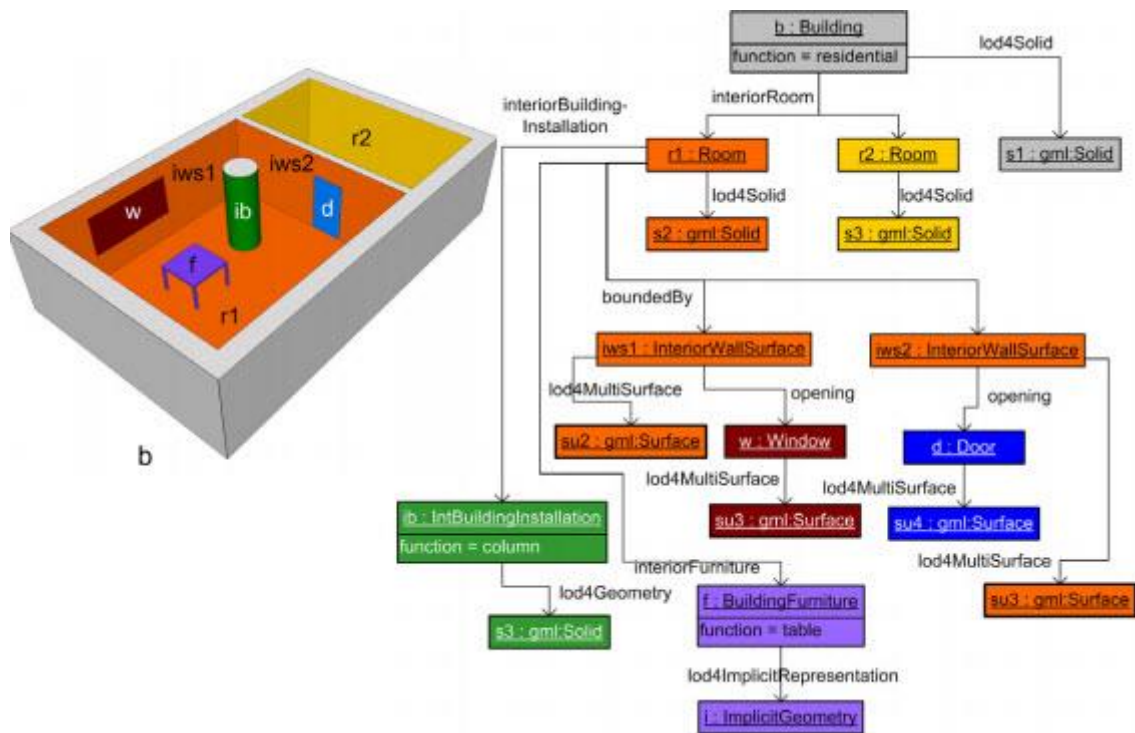
Şekil 5. 13 LoD3’de bina temsili: (a) geometrik temsil, (b) UML örnek diyagramı [7]



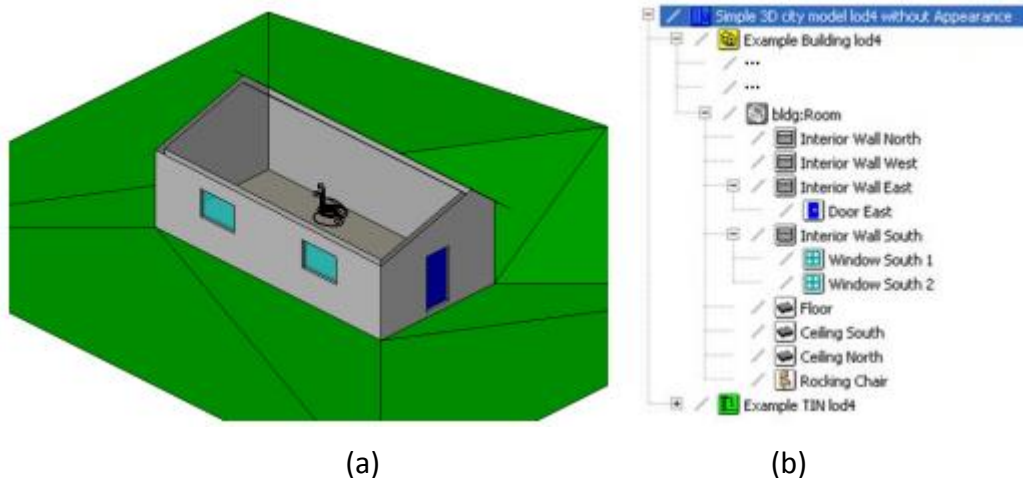
Şekil 5.14 LoD3’de CityGML bina modeli: (a) 3B geometri ve (b) model hiyerarşisi [9].

5.1.5 Ayrıntı Düzeyi 4 (LoD4)

LoD4'te, LoD3'e ek olarak binaların iç yapıları dikkate alınmaktadır. Şekil 5.15'teki örnekte, b binası (LoD3'te olduğu gibi katı model $s1$ ile temsil edilen şeklin dış yüzeyi), $r1$ ve $r2$ odalarına sahiptir. Her odanın geometrisi bir katı ($s2$, $s3$) ile temsil edilmektedir. $r1$ odası, (w , d) açıklıklarına sahip olan ($iws1$, $iws2$, ...) iç duvar yüzeyi detayları ile sınırlandırılmaktadır [7].



Şekil 5. 15 LoD4'te bina temsili: geometrik temsili ve UML örnek diyagramı [7]



Şekil 5. 16 LoD4'te CityGML bina modeli örneği: (a) 3B geometri ve (b) model hiyerarşisi. Çatı yüzeyleri, hem iç sınır yüzeyleri hem de bina mobilyasını görselleştirmek için gösterilmemiştir [9].

LoD4'de bina, iç kısmının temsili ile tamamlanmaktadır (Şekil 5.16). Örnekteki modelde, sallanan sandalye (*bldg:BuildingFurniture*) ile donatılmış bir oda (*bldg:Room*) bulunmaktadır. Oda, geometileri odanın LoD4 katı geometrisi tarafından referanslanan (*xlink:href*) geometrilerin (odanın *bldg:boundedBy* özniteliği aracılığıyla ilişkilendirilen *bldg:InteriorWallSurface*, *bldg:FloorSurface*, *bldg:CeilingSurface*) iç sınır yüzeyleri tarafından sınırlandırılmaktadır [9].

5.2 INSPIRE ve CityGML Bina Modelleri Arasındaki Farklar

INSPIRE birlikte çalışabilir şekilde kamu sektörü verilerini sağlayarak Avrupa mekansal veri altyapısının oluşturulmasını hedefleyen Avrupa Komisyonu'nun bir inisiyatifidir. 34 kavram için yaygın semantik modeller (idari birimler, binalar, ulaştırma şebekesi, arazi kullanımı, jeoloji, vb.), birlikte çalışabilen verilerin değişimini sağlamak için Tematik Çalışma Grubu (TWG) tarafından oluşturulmaktadır. INSPIRE Bina Tematik Çalışma Grubu (TWG BU) CityGML için aşağıdaki değişiklikleri uygulamaya koymaktadır [12]:

CityGML, INSPIRE için aşağıdaki gibi uyarlanmıştır:

- Örnek 1: CityGML'de genel olarak uyumlaştırılmış binaların sınıflandırılmasına ilişkin öznitelikler, (örneğin, Alman Kadastro'su'na verilen karakter katarı (*string*) veya kod listesi olarak ilan edilen) INSPIRE Bina Tematik Çalışma Grubu tarafından elde edilen kullanıcı gereksinimleri ile daha iyi uyumlaştırılacak şekilde yeniden yapılandırılmaktadır.
- Örnek 2: Karşılıklı tema tutarlılığı sağlamak için, bütün INSPIRE teması için ortak öznitelikler (veri setindeki tanımlayıcılar ve zamansal öznitelikler) CityGML tarafından sağlananlar yerine, INSPIRE Genel Kavramsal Modeli (INSPIRE Generic Conceptual Model) tarafından sağlanan ad ve tanımlamaları kullanılmaktadır.

CityGML, INSPIRE için aşağıdaki gibi basitleştirilmiştir:

- Örnek 1: CityGML'de *Building* ve *BuildingPart* arasındaki birleştirme hiyerarşisi yinelemelidir yani bir parça parçalara sahip olabilir fakat INSPIRE'da bir parça parçalara sahip olamaz (CityGML UML diyagramında birleştirme hiyerarşisi *AbstractBuilding* sınıfında başlar ve hem *Building* hem de *BuildingPart*

tarafından devralınır, oysa ki INSPIRE'da birleştirme hiyerarşisi *Buildings*'de başlar) .

- Örnek 2: Görünüş için model büyük ölçüde basitleştirilmiştir. Yalnızca bir binanın doğal görünüşünü modellemek için tek uygun model olan parametrikleştirilmiş doku (*Parameterized Texture*) modeli tutulmaktadır. Bununla birlikte CityGML, kızılötesi radyasyonu veya gürültü kirliliği gibi fiziksel olguların etkisinin de modellenmesine de olanak tanır).

INSPIRE Bina Tematik Çalışma Grubu, CityGML içeriğini aşağıdaki gibi zenginleştirmektedir:

- Örnek 1: Bazı detay türleri, Bina Birimi (*BuildingUnit*) ve birçok öznitelikler (Konut Sayısı (*numberOfDwellings*), Yapının Durumu (*ConditionOfConstruction*)) INSPIRE kapsamında çevre politikaları ile ilgili kullanıcı gereksinimlerini karşılamak için TWG BU tarafından modele eklenmiştir.
- Örnek 2: CityGML bina modelinin iki uzantısı *extended3D* profilinde gösterilmektedir: Tematik Sınır Yüzeyi (*BoundarySurfaces*), Çatı malzemesi (*materialOfRoof*) ve Cephe Malzemesi (*materialOfFacade*) özniteliklerine sahiptir.
- Örnek 3: *Extended2D* ve *Extended3D* profillerinde, CityGML modelinin kapsamı, rüzgar değirmenleri ve antenler gibi bina olarak dikkate alınmayan yapıları içerecek şekilde genişletilmiştir.
- Örnek 4: CityGML, LoD1 temsilinin en üst referans ve LoD1-4 temsili için en alt referans değerinin doğrudan depolanmasıyla genişletilmektedir. Bunun yanında, LoD1/LoD2 taban alanının başvuru (referans) yaptığı bina parçası doğrudan verilmektedir.
- Örnek 5: CityGML'in iç mekan modeli, Bina Birimleri (*BuildingUnits*) tarafından zenginleştirilmektedir.

6.1 Veriler

Uygulamada veri olarak, İstanbul'da yer alan İstanbul Erkek Lisesi, Çırağan Sarayı, Ayasofya ve Atatürk Kitaplığı'na ilişkin Sketchup programında modellenmiş dört adet 3B bina modeli seçilmiştir. Verileri seçilirken, 3B modellerin özellikle farklı tipte bina ve bina parçaları içermelerine dikkat edilmiştir.

6.2 Yazılım

Ticari veya akademik araçlar, görüntüleyiciler veya servisler, veri alış-verişi sağlayan arayüzler ile CityGML'i desteklemektedir. Çizelge 6.1'de ticari yazılımların yanında CityGML'i destekleyen ücretsiz araç ve hizmetler listelenmiştir [46]. Özellikle karmaşık nesne yönelimli modeller, semantikler, tek bir detayın farklı ayrıntı düzeylerinde temsili ve parçalar içinde nesne ayrıştırma ile ilgili CityGML (1.0 versiyonu) işlevlerini desteklemek için talep edilen araçlar vardır. 3D City DB, FME, Bentley Maps SS2 (sadece çoklu yüzeyleri destekler) ve LoD1-LoD3 ve bina, bitki örtüsü, su kütlesi, kent donatıları, tünel, köprü detayları için CityGRID gibi yazılımlar örnek gösterilebilir [7].

Farklı ayrıntı düzeyinde 3B kent modellemeye olanak sağlayan yazılımların, modelin hem geometrik hem de semantik olarak yapılandırılmasına olanak tanınması gerekmektedir. Bu bağlamda, deneme imkanı bulduğumuz bazı yazılımlar incelenmiş yazılımlar incelenmiş ve amaca en uygun yazılım olan CityGRID™ kullanılmıştır.

Çizelge 6.1 CityGML’i destekleyen araçlar ve hizmetler (7 ve 46’dan uyarlanmıştır)

Araç	Kuruluş/ Şirket
3B CityGML görüntüleyicileri	
Aristoteles 3D	Univ. of Bonn
CityGML-Toolchain	Univ. of Appl. Sci. Gelsenkirchen
FZKViewer	KIT Karlsruhe
BS Contact Geo	Bitmanagement Software GmbH
Cityvu	
FME Data Inspector	Safe Software Inc.
Tridicon CityDiscoverer	Hexagon
Viewtec Terrainview	ViewTec Inc.
RhinoTerrain/Rhino	SARL RhinoTerrain
TerrainView	TerrainView
GEORES CityGML SpiderViewer	GEORES
IN3d Visulation Engine	Galdos systems Inc.
CityGML sağlayan Web özellikli hizmet araçları	
Go Publisher WFS	Snowflake Software Ltd.
XtraServer	Interactive instruments
CityServer3D	Fraunhofer
SupportGIS WFS-T	CPA Geo-Information
INscape WFS-T	Galdos systems Inc.
CityGML verilerini kontrol etmek için araçlar	
QS-City 3D	Univ. of Appl. Sci. Stuttgart
CityDoktor Validator	Univ. of Appl. Sci. Stuttgart
Veritabanları	
Oracle Spatial 11g	Oracle Corp.
3DCityDB	Technische Univ. Berlin, IGG
Sınıf kütüphanesi/API	
Citygml4j	Technische Univ. Berlin, IGG
CityGML’i destekleyen veri dönüştürme araçları	
FME	Safe Software Inc.
Libcitygml	
CityGML – CBS dönüşümü	
Bentley Map SS2	Bentley Systems, Inc.
CityGML sağlayan veri toplama araçları	
CityGRID	UVM Systems GmbH
Tridicon CityModeller	Hexagon
novaFactory 3D	M.O.S.S. GmbH
MBRCR (Model-Based Roof Classification and Reconstruction system)	Univ. of Bonn, IGG
CityGML Plugins Sketchup™-Sketchsoftware	GEORES
CityGML veri toplama hizmetleri	
	Virtualcitysystems
	COWI
	Blom ASA
	Bimtas
	GTA Geoinformatik
	M.O.S.S. GmbH
	Vectuel

Ayrıca kullanılan veriler SketchUp (.skp) dosya formatında olduğundan ve aynı zamanda verileri birimlerine ayırabilmek için dönüşüm işlemi gerektiğinden SketchUp, diğer taraftan modeli CityGML standardına dönüştürme öncesi gerekli geometrik düzenlemeleri yapmak için Autodesk 3Ds Max ve ayrıntı düzeylerini görüntülemek için LandXplorer Viewer yazılımları kullanılmıştır.

6.2.1 CityGRID

CityGRID™, 3B kent modellerinin oluşturulması ve yönetimi için etkin ve uygun maliyetli bir çözüm olarak kent çapında veya belirli bölgeye özgü, proje yönelimli modellerin uzun dönemde sürdürülebilirliği için uygun, uygulamada kullanıcı ile uzun süren iş birliği sonucunda geliştirilen ve ayrıca uyarlanabilen ara yüzü sayesinde, mevcut iş akışları ve sistem ortamlarına en uygun şekilde entegre edilebilen bir yazılımdır [42].

CityGRID yazılımı, Viyana/Avusturya merkezli UVM Systems GmbH tarafından geliştirilmektedir. UVM Systems GmbH, CityGRID ürün yelpazesi ile 3B kent modellerine çözümler getiren, modelleme, yönetim ve 3B kent modellerinin kullanımı için yazılımlar geliştiren, gerçek zamanlı 3B görselleştirmeler üzerine uzmanlaşmış yenilikçi bir coğrafi bilgi şirkettir [43].

Ayrıca yazılım, 3B kent modeli ile ilgili olarak; kent verilerinin CityGML, CAD ve CBS ortamlarında geçerli diğer formatlara aktarımı ve hava fotoğraflarından bina dokularının foto gerçekçi dokulandırılması işlemini gerçekleştirmektedir. Mevcut kent modellerinin yönetilebilir, güncellenebilir ve sürdürülebilir duruma getirilmesinde, 3B binalarda çatı ve cephe üzerindeki nesnelerin modellenerek bina ayrıntısının artırılmasında ve mevcut bina modellerinin 3B sunumlar ve analizlere uygun yapıya kavuşturulmasında etkindir [44].

CityGRID™ yazılımı, *Manager, Administrator, Modeler, Builder ve Scout* modüllerinden oluşmaktadır [45].

CityGRID Manager, kent modelinin veri tabanı desteği ile kolay yönetilebilir ve düzenlenebilir halde saklanmasını ve 3B bina modellerinin mobil haritalama (coğrafi referanslı veri toplama) ile toplanan veriler ve hava fotoğrafları ile otomatik kaplanması işlemlerini gerçekleştirmektedir.

CityGRID Administrator, veri yönetimi için yardımcı programdır. CAD verileri, standart bir XML formatına dönüştürülür ve veri tabanı içerisine dahil edilebilir. Ayrıca veriler, çeşitli formatlarda aktarılabilir. Kent modeli verilerinin yönetimi altında binalara otomatik doku giydirmeye gibi belirli fonksiyonlar, bu program tarafından yönetilmektedir [45].

CityGRID Modeler, 3B binaları modelleme, detaylandırma ve kaplama için kullanılmaktadır ve 3B kenar çizgileri ile modellerin etkin bir şekilde düzeltilmesini sağlamaktadır. Modeler, bir veri tabanı içerisinde (*CityGRID Manager*) veya standart XML formatındaki dosyalarda tutulan veri ile bağlantılı olarak kullanılabilir. *Modeler*, *Inspector*, *Editor* ve *Texture* olmak üzere üç modül kullanılabilir. *CityGRID Modeler*'in tam versiyonu, bütün modüllerin birleşiminden oluşur [45]. *CityGRID™* yazılımının Modeler ve Builder modülü, Autodesk 3D max yazılımı üzerinde eklenti (plug-in) olarak çalışmaktadır.

CityGRID Builder, mimarlara, farklı planlama seçeneklerine ilişkin görselleştirmelerin ve *CityGRID Scout* için en uygunlaştırılmış sahneleri oluşturmaları için mevcut kent modelleri içerisine kendi planlarını entegre etme olanağı sağlamaktadır [45].

CityGRID Scout, binalar ve bitki örtüsü ile birlikte yolların son derece gerçekçi görünüşe sahip olduğu sanal gerçeklik görselleştirmeleri için etkileşimli 3B gerçek zamanlı görüntüleyicidir. Gerçekçi kapsamda, tasarlanmış nesnelerin etkisinin simülasyonuna olanak tanır [45].

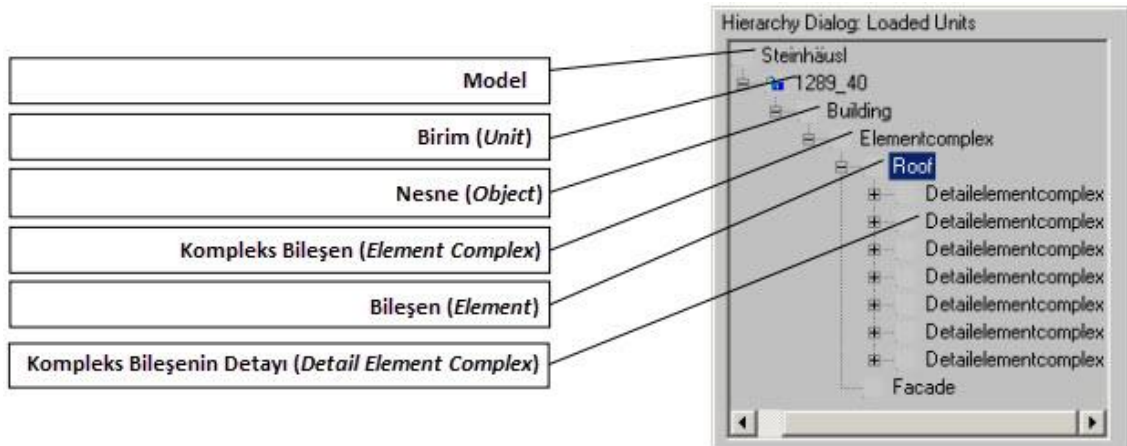
6.2.1.1 CityGRID’de Bina Semantiği

CityGRID’de binalar, mantıksal olarak gruplandırılmıştır. Bina komplekslerinin çok detaylı hiyerarşisi, *CityGRID Modeler*’da ‘*Loaded Units*’ penceresinde gösterilmektedir (Şekil 6.1).

- Model

Model, herhangi bir sayıda Birim (*Unit*)’den oluşan mantıksal gruptur. Başka bir ifadeyle, her Birim (*Unit*) herhangi bir sayıda *Model*’e aittir. Bina toplulukları şeklinde *Model*’in veri tabanı versiyonu ile çalışmasını kolaylaştıran Birim (*Unit*)’ler, kolaylıkla

Model oluşturmak için gruplandırılabilir. Bir *Model*'in silinmesi, Birim (*Unit*) üzerinde bir sorun oluşturmaz.



Şekil 6. 1 CityGRID’de bina modeli hiyerarşisi [45]

- Birim (*Unit*)

Birim (*Unit*), görüntünün yönetsel açıdan binaların oluşturduğu gruptur. Örneğin bir mülkiyete ait olan bütün binalardan oluşur. Bir Birim (*Unit*)’in en az bir Nesne (*Object*)’ye sahip olması gerekir.

- Nesne (*Object*)

Nesne (*Object*) tek bir binadır. Her Nesne (*Object*) tam olarak bir Birim (*Unit*)’in parçasıdır. Her Nesne (*Object*)’nin en az bir Kompleks Bileşen (*Element Complex*)’i vardır. Birim (*Unit*)’e ait Nesne (*Object*)’ler genelde geometrik olarak ayrılır. Nesne (*Object*) önceden tanımlanan Nesne (*Object*) sınıflarına (bina, köprü, yeraltı gibi) sahiptir.

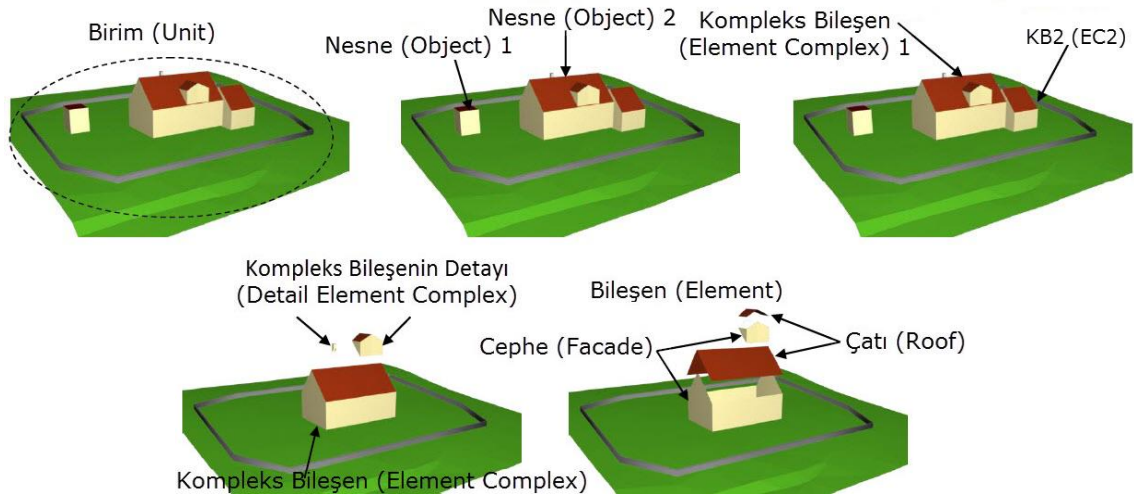
- Kompleks Bileşen (*Element Complex*)

Kompleks Bileşen (*Element Complex*), tamamı bir katıyı tanımlayan bileşenlerden oluşan gruptur. Her Kompleks Bileşen (*Element Complex*) tam olarak bir Nesne (*Object*)’nin parçasıdır. Bir Kompleks Bileşen (*Element Complex*), bir Nesne (*Object*)’nin katı (solid) olarak tanımlanan esas elemanı olabileceği gibi Kompleks Bileşen (*Element Complex*)’e bağlı eklentilerden (çatı penceresi gibi) yani Kompleks Bileşenin Detayı (*Detail Element Complex*)’ndan da oluşabilir [45].

- Bileşen (*Element*)

Bileşen (*Element*), Kompleks Bileşen (*Element Complex*)'in parçasıdır ve çatı, cephe, çatı payı (çıkıntı) veya taban gibi parçaların yüzeylerini tanımlar. Çatı ve/veya tavan bileşenleri Kompleks Bileşenin Detayı (*Detail Element Complex*)'na sahip olabilir. Her Kompleks Bileşenin Detayı (*Detail Element Complex*) en az bir Bileşen (*Element*)'e sahip olmalıdır ve Bileşen (*Element*)'ler geometrik bilgileri taşır [45].

Bina modellerinin semantik modellemeyi sağlayan hiyerarşik yapısı Şekil 6.2'de bir örnekle gösterilmektedir. Burada semantik yapı, binanın çatısına ait nesneleri birer "detay" olarak kaydetmeyi sağlamaktadır. Örneğin ana çatı, herhangi bir değişikliğe uğradığında detaylar otomatik olarak yeni ana çatıya göre şekillenirler. Böylelikle, semantik model yapısı, binanın geometrisini oluşturmada kullanılabilir. [45].

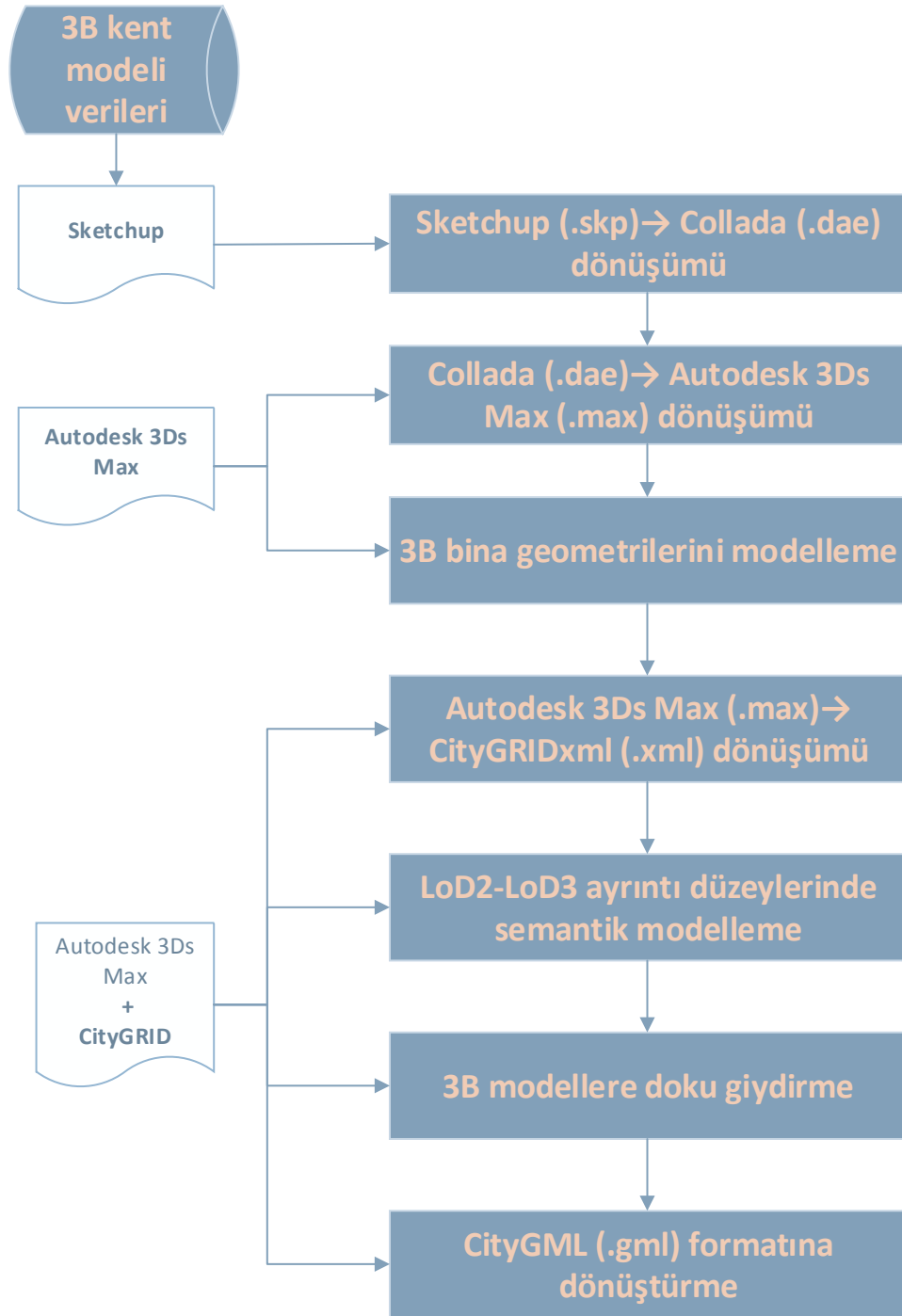


Şekil 6. 2 CityGRID' de bina semantiği [45]

6.2.1.2 CityGRID'de Farklı Ayrıntı Düzeylerini (LoD) Modelleme

CityGRID™'de bina modelleri (ana veriye bağlı olarak) mümkün olan en yüksek ayrıntı düzeyinde modellenmektedir. Alt ayrıntı düzeyleri model veri tabanına (Oracle veya SQL Server) aktarıldığında sistem tarafından otomatik olarak oluşturulmaktadır. Kullanıcı istediğinde örneğin LoD3 seviyesindeki binaya ait LoD2, LoD1, LoD0 ayrıntı düzeyini çağırabilir ya da istenen bir formata (örn. CityGML) dönüştürülebilir. Yazılımın sahip olduğu mevcut fonksiyonlar, LoD4 modeller oluşturmaya uygun değildir. Modellerin CBS ile entegrasyonu ise her binanın taşınması zorunlu olan birincil anahtar (*primary key*) ile sağlanmaktadır [43].

6.3 Gerçekleştirme

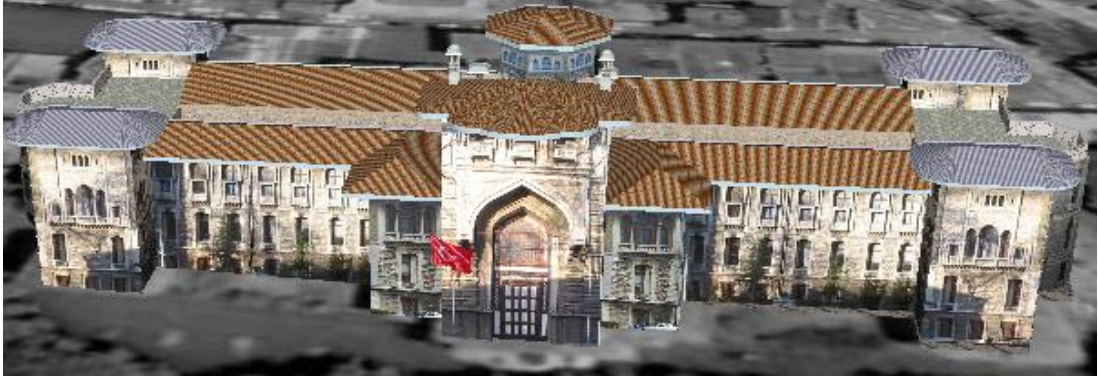


Şekil 6. 3 Uygulama işlem adımları

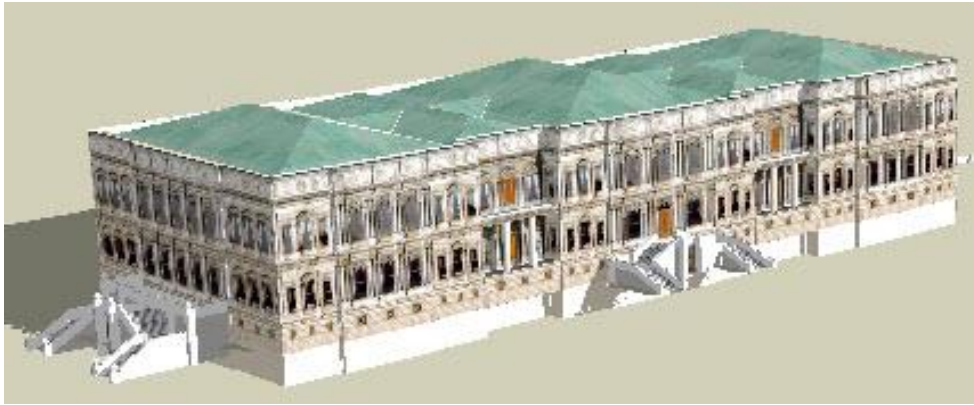
6.3.1 Format Dönüşümü

Uygulamada ilk olarak Sketchup (.skp) formatındaki 3B örnek kent modeline ait 3B bina verileri, COLLADA (.dae) formatına dönüştürülmüştür. COLLADA formatının tercih edilmesinin nedeni, 3B bina modellerinin Autodesk 3Ds Max yazılımına parçalar

halinde aktarılmasını sağlamasıdır. Bu durum daha sonraki geometrik modelleme aşamasını kolaylaştırmaktadır. Ayrıca 3B modellere ilişkin dokular bozulmamaktadır. Şekil 6.4, Şekil 6.5, Şekil 6.6 ve Şekil 6.7’de 3B modellerin ilk hali (.skp formatında) görülmektedir.



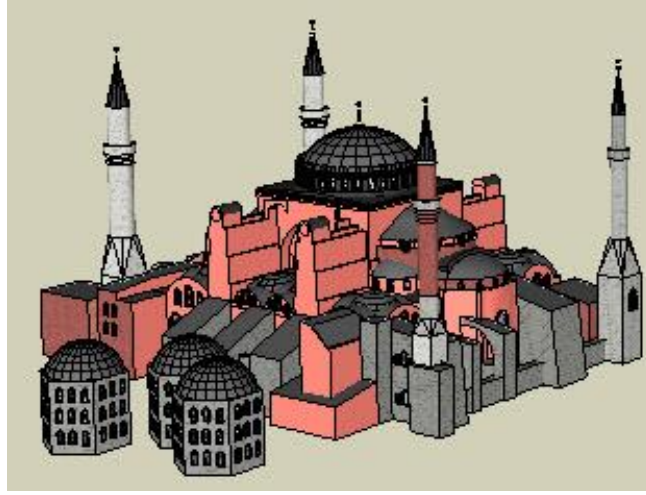
Şekil 6. 4 İstanbul Erkek Lisesi’nin Sketchup formatındaki 3B modeli



Şekil 6. 5 Çırağan Sarayı’nın Sketchup formatındaki 3B modeli



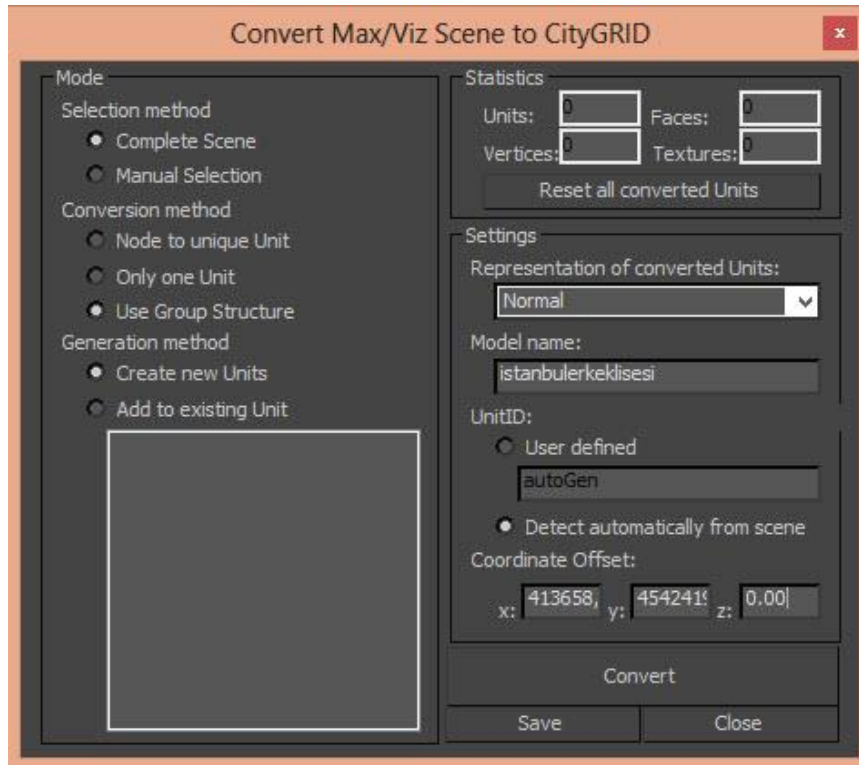
Şekil 6. 6 Atatürk Kitaplığı’nın Sketchup formatındaki 3B modeli



Şekil 6. 7 Ayasofya'nın Sketchup formatındaki 3B Modeli

Geometrik olarak gruplandırma ve doku giydirme kontrolü yapıldıktan sonra Autodesk 3Ds Max yazılımının kendi formatı olan 3ds Max (.max) formatına dönüştürülmüştür.

3B modellerin Birim (*Unit*) bazında gruplandırılmasından sonra CityGRIDxml'e dönüştürme aşamasına geçilmektedir. Bu dönüşüm işlemi, CityGRID Tools'da *BLOB Conversion Manager* alt menüsüyle yapılmaktadır. Dönüşüm arayüzü Şekil 6.8'de gösterilmektedir.

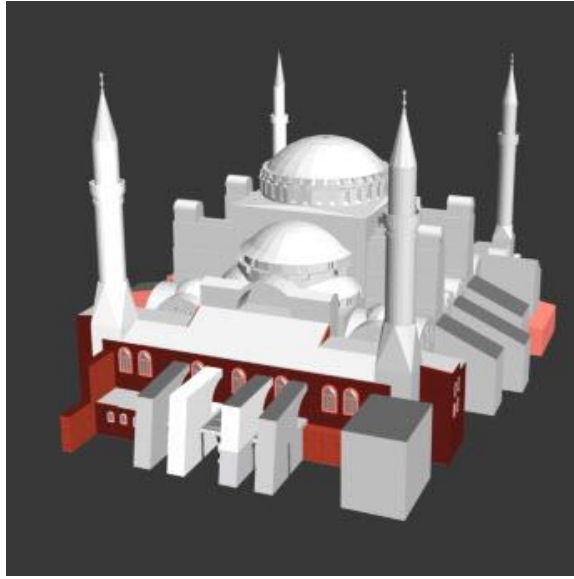


Şekil 6. 8 İstanbul Erkek Lisesi için CityGRIDxml dönüştürme



Şekil 6. 9 Atatürk Kitaplığı CityGRIDxml dosyası

Şekil 6.9 ve 6.10’da 3ds Max formatından CityGRIDxml dosyasına dönüştürülen Atatürk Kitaplığı ve Ayasofya 3B bina verisi görülmektedir.



Şekil 6. 10 Ayasofya CityGRIDxml dosyası

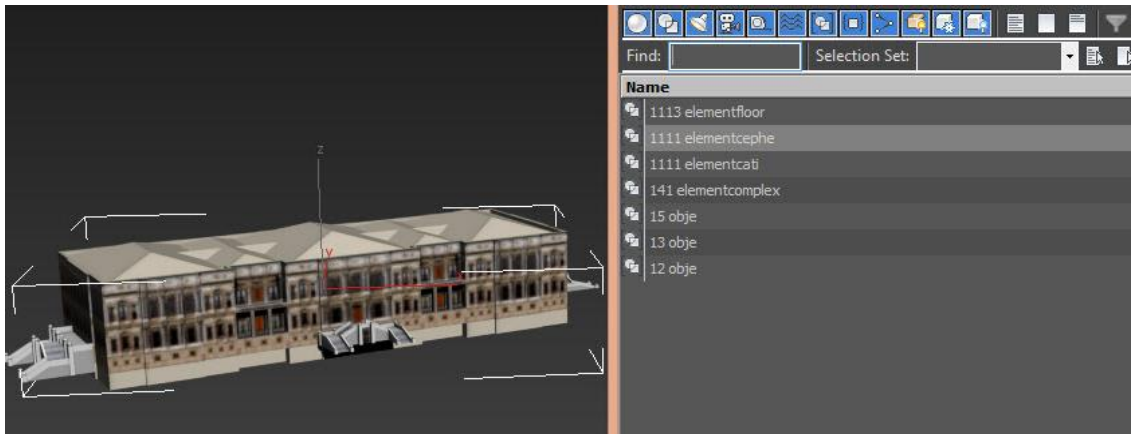
6.3.2 Koordinat Dönüşümü ve Boyutlandırma

3B bina modelleri, İstanbul Büyükşehir Belediyesi kent rehberinde yer alan bina boyutlarıyla (en ve boy) karşılaştırıldığında aynı değerlere sahip olmadığından gerçek değerlerine göre ölçeklendirilmiştir. Yalnızca Ayasofya 3B modeli için dönüklük problemi olduğundan yine kent rehberi dikkate alınarak gerçek değerine göre döndürülmüştür.

Sonrasında 3B bina modelleri, İstanbul Büyükşehir Belediyesi kent rehberinin GK Orta Meridyen 30° (ITRF96) koordinat sisteminde tanımlı haritalarından elde edilen sağa ve yukarı değerleri ile CityGRID'deki lokal dik koordinatlarının farkından koordinat kaydırma değerleri hesaplanarak gerçek konumlarına oturtulmuştur. Kaydırma değerleri CityGRIDxml dönüşümü esnasında, dönüşümü sağlayan araç üzerinden girilmektedir.

6.3.3 Geometrik Modelleme ve İşleme

CityGML standardına uygun semantik yapı oluşturabilmek amacıyla ön işlem olarak geometrik gruplandırma işlemine geçilmiştir. Modeller oluşturulurken CityGML standardı göz önüne alınmadığından bu aşamada bina, geometrik olarak cephe ve taban gibi parçalarına ayrılır. Sonrasında bir bütün (örn. kubbe) oluşturan her parça kendi arasında geometrik olarak birleştirilmiştir. Gruplandırma işlemi, bina ve binaya ait parçaların ilk basamağı olan bileşen (cephe, çatı, taban ve katı) şeklinde birleştirilerek başlanmıştır. Örneğin, Çırağan Sarayı bir bina ve binaya ait ve bitişik detaydan (merdivenler) oluştuğu için bu örnek için bir bina ve dört bina tesisatı (*BuildingInstallation*) biçiminde bir semantik yapı düşünülmüştür. Bileşenler, kendi aralarında gruplandırılarak sırasıyla *Element Complex*, *Object* ve *Unit* şeklinde CityGRID bina semantiği oluşturulmuştur. Şekil 6.11'de Çırağan Sarayına ilişkin gruplandırma işlemi görülmektedir. Gruplandırma işleminin her zaman son hali bina ve binaya ait (bitişik veya ayırık) tüm detayları kapsayan *Unit* grubudur.

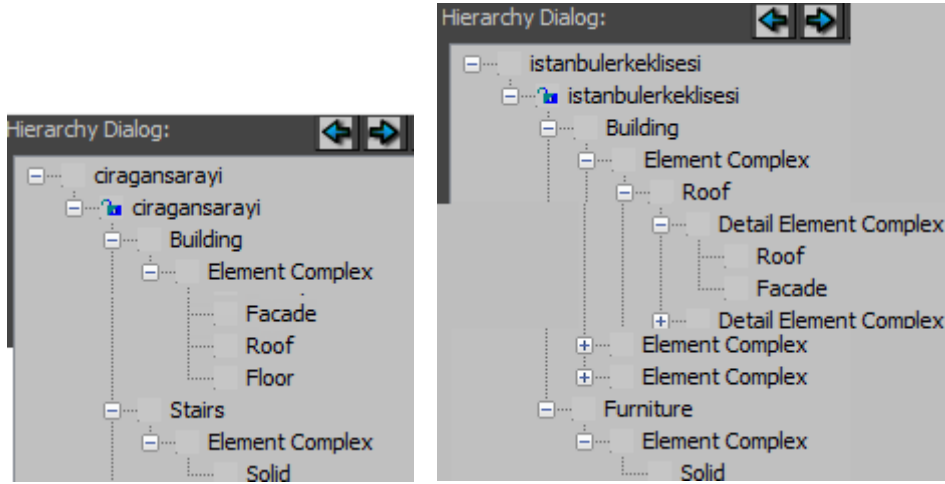


Şekil 6. 11 Çırağan Sarayı geometrik gruplandırma aşamaları

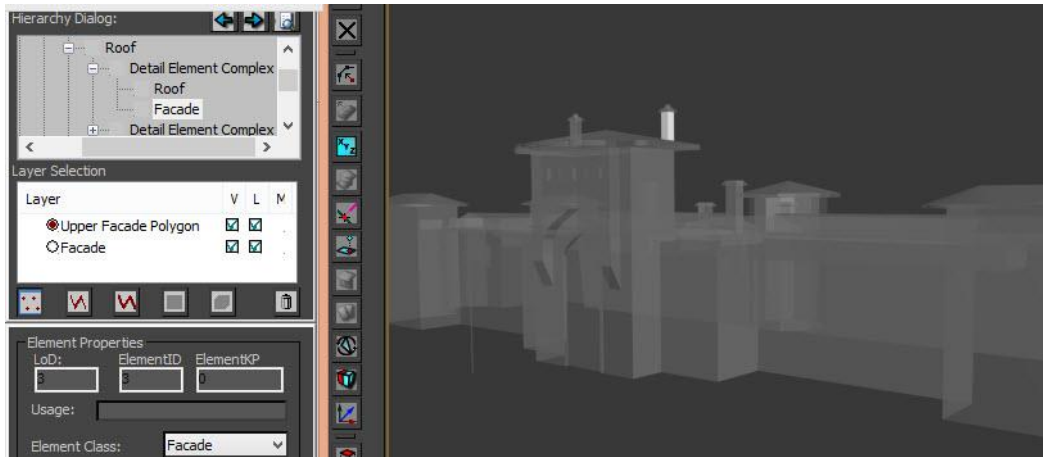
6.3.4 LoD2 ve LoD3 Ayrıntı Düzeylerinde Semantik Modelleme

3B modeller, CityGRID’de xml formatına dönüştürüldükten sonra geometrik olarak gruplandırılan bina bileşenlerinin semantik yapısı, CityGRID mantığına göre hiyerarşik olarak oluşturulmuştur (Şekil 6.12).

Ayrıca örnek veriler için oluşturulan semantik yapı ve bu yapının aktif olan parçası gösterilebilmektedir. Aynı zamanda *Element Properties* penceresinde bileşenlerin hangi ayrıntı düzeyine ait olduğu da görülebilmektedir. Çatıya ait birim, çatı detayı olduğundan LoD3 ayrıntı düzeyindedir. Şekil 6.13’te İstanbul Erkek Lisesi için çatı detayın cephe bileşeninin semantik yapısı ve bu yapının ait olduğu LoD3 ayrıntı düzeyi görülmektedir.

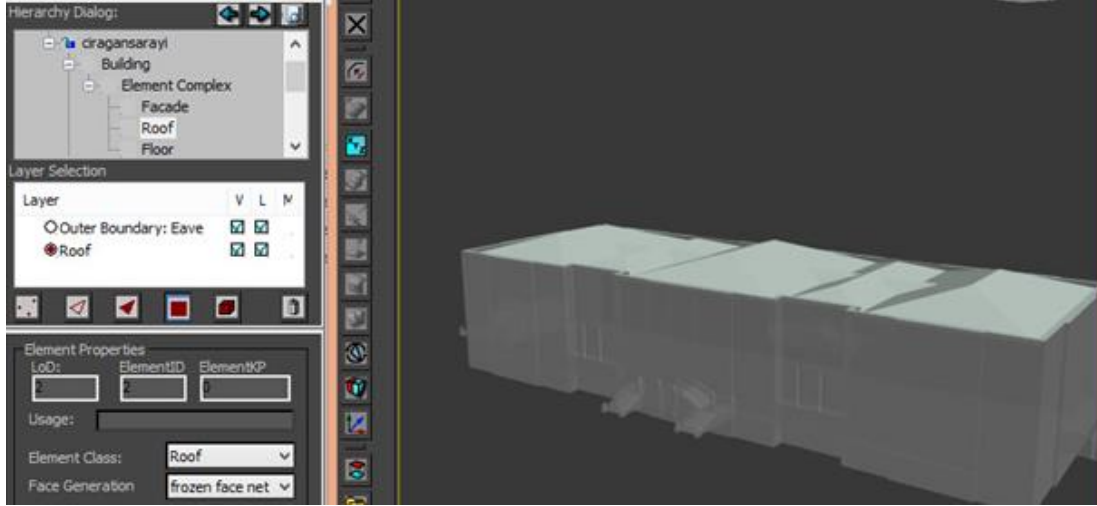


Şekil 6. 12 3B örnek modeller için CityGRID’de bina ve bileşenlerine ilişkin hiyerarşik semantik yapı



Şekil 6. 13 Çatı detayının (cephe) semantiği ve LoD3 temsili

Benzer şekilde, Çırağan Sarayı örneği için CityGML standardında Çatı Yüzeyi'ne (*RoofSurface*) karşılık gelen çatı bileşeni, LoD2 ayrıntı düzeyindedir. Bu nedenle, Şekil 6.14'te, çatı bileşeninin semantik yapısı ve bileşen LoD2 ayrıntı düzeyinde olduğundan yazılım ara yüzünde bu ayrıntı düzeyi temsili görülmektedir.



Şekil 6. 14 Bina çatısının semantik yapısı ve LoD2 temsili

6.3.5 3B Modellere Doku Giydirme

3B Modellerin semantik yapısı oluşturulduktan sonra CityGRID yazılımında bulunan doku giydirme araçları kullanılmıştır. İki farklı yöntem ile bu işlem gerçekleştirilebilmektedir. Birinci yöntemde, COLLADA dönüştürme sırasında oluşan dokulara ilişkin görüntü dosyalarından bina ve bina nesnelere ait görüntüler manuel olarak giydirilebilmektedir. İkinci yöntemde ise, CityGRID yazılımında bulunan doku seçenekleri kullanılmaktadır. 3B modellerin oluşturulması esnasında farklı doku giydirme yöntemleri kullanıldığından bina veya bina nesnesi için görüntü dosyasından doku giydirmek yetersiz kalmaktadır. Bu durumda yazılımın sunduğu doku seçenekleri kullanılmaktadır. Şekil 6.15'te bu durum Çırağan Sarayı örneğinde gösterilmektedir. Modelin cephe fotoğraflarından bazıları dönüştürme esnasında korunmuştur. Korunmayan cephe fotoğrafları ilk yöntemle, çatı fotoğrafları ve bina detayları ikinci yöntemle kaplanmıştır. Ancak yine de örnekte görüldüğü gibi binanın sağ ve sol köşe cephelerine giydirilen dokular, modeller oluşturulurken farklı doku giydirme teknikleri kullanıldığından cepheye istenen şekilde oturmamıştır.



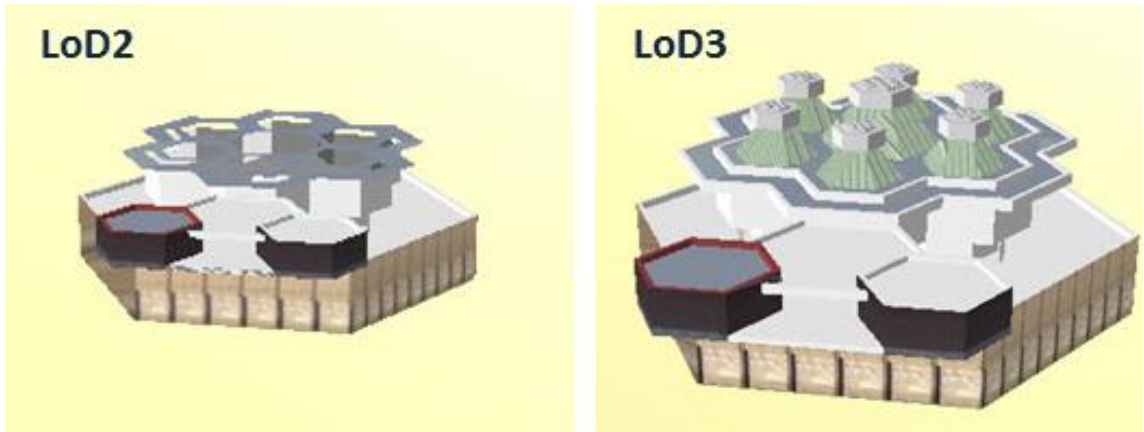
Şekil 6. 15 Doku giydirilmiş Çırağan Sarayı örneği

6.3.6 CityGML Formatına Dönüşüm

3B bina modelleri, yukarıda belirtilen işlemler uygulandıktan sonra CityGRID yazılımıyla CityGML formatına dönüştürülmüştür (Bkz. Bölüm 6.5).

6.4 Bulgular

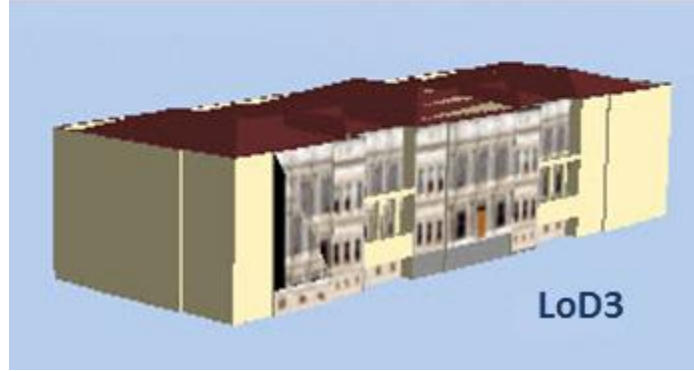
CityGML dosya formatına dönüştürülen 3B bina modelleri, CityGML dosya formatındaki verileri açmak için yaygın olarak kullanılan LandXplorer Viewer yazılımı'nda görüntülenmiştir. Şekil 6.16 ve Şekil 6.17'de uygulama için seçilen 3B örnek modellerden Atatürk Kitaplığı ve Çırağan Sarayı, LandXplorer Viewer yazılımı'nda LoD2 ve LoD3 ayrıntı düzeylerinde gösterilmektedir.



Şekil 6. 16 Atatürk Kitaplığı'nın LoD2 ve LoD3 ayrıntı düzeyinde temsili



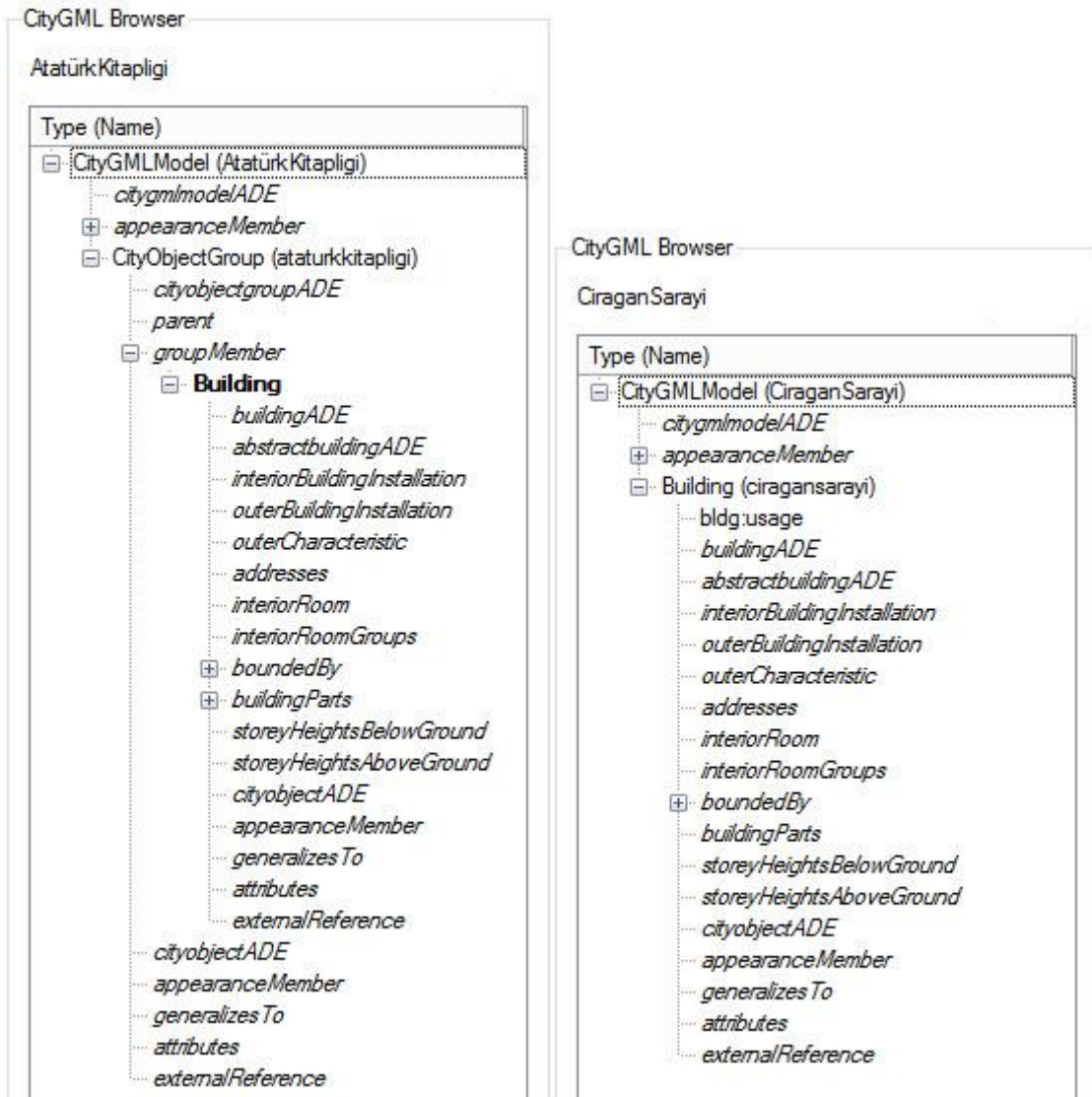
(a)



(b)

Şekil 6. 17 Çırağan Sarayı'nın LoD2 (a) ve LoD3 (b) ayrıntı düzeyinde temsili

Aynı zamanda 3B bina modellerinin CityGML öznetelik türleri de LandXplorer Viewer yazılımında görüntülenebilmektedir. Şekil 6.18'de Atatürk Kitaplığı ve Çırağan Sarayı, 3B modeller için CityGML öznetelik türleri gösterilmektedir.



Şekil 6. 18 Atatürk Kitaplığı ve Çırağan Sarayı için CityGML öznelik türleri

6.5 Sorunlar

Uygulamada kullanılan 3B modeller, görselleştirme amaçlı oluşturulduğundan ve farklı formatlarda olması nedeniyle bu modelleri CityGML standardına uygun olarak modelleme sürecinde özellikle format dönüşümü, doku giydirme ve geometrik olarak gruplandırma aşamalarında bir takım problemlerle karşılaşmıştır.

İlk olarak Sketchup dosya formatındaki (.skp) modelleri diğer formatlara dönüştürme çalışmasında modellerin tekli gruplar halinde dönüşmesi sorunu ile karşılaşmıştır. Bu durum, geometrik olarak düzenlemeye imkan vermeyeceğinden bazı dönüşüm formatları (.fbx, .dxf, .vrml, .3ds, .dae) denenmiştir. Bu formatlardan istenen sonucu

veren COLLADA (.dae) tercih edilmiştir. Böylece 3B modeller, tek nesne olarak dönüşmemiştir yani geometrik olarak gruplandırmaya uygundur.

Sonrasında, geometrik olarak gruplandırma işleminde 3B modellerin CityGRID hiyerarşisine uygun hale getirilebilmesi için *Element Complex* sınıfında en az iki bileşenin (cephe, taban ya da çatı) olması gerektiğinden; modeller, bu mantık doğrultusunda gruplandırılmaya çalışılmıştır. Bu işlem sırasında Çırağan Sarayı örneğinde olduğu gibi dönüşüm sonucu ya da modelleme hatası olarak cephenin bir kısmı taban ile bütünleşik yapıda olduğu için önce birbirinden ayrılmış, daha sonra cephe ve taban olarak kendi aralarında gruplandırılmıştır. Atatürk kitaplığında cephe ve çatı detayının parçalı modellenmemesinden kaynaklanan semantik yapıya uymama problemi vardır: Bu işlemde bina parçasına ait çatı detayı oluşturulamamış ve cephe bileşeni olarak gruplandırılmıştır. Benzer sorun Ayasofya örneğinde de görülmüştür. Özellikle 3B modelin oldukça karmaşık olması geometrik modellemeyi güçleştirmekte ve zaman alıcı hale getirmektedir. Ayrıca bu örnek modelde Kompleks Bileşen (*Element Complex*)'ler arasında ortak cephenin olması diğer bir problemdir. Bu sorunu CityGRID yazılımı CityGML dönüşümü sırasında ortak yüzeyleri dönüşüme almayarak düzeltmektedir. Öte yandan yine Ayasofya 3B modeli için taban bileşeni oluşturulmasında sorun olmuştur. Bir taban sadece bir Kompleks Bileşen (*Element Complex*)'e ait olmalı mantığına uymayan durumlar meydana gelmiştir. Bu sorunda Autodesk 3Ds Max'de her Element Complex'e bir taban ait olacak şekilde geometrisinin düzenlenmesiyle ortadan kaldırılmıştır.

Ayasofya örneğinde görülen farklı bir sorun ise LoD3 ayrıntı düzeyine ait pencere açıklıkları CityGML standardında cephe detayı iken, CityGRID'de bu detaya karşılık gelen bir bileşen detayı (*Detail element complex*) yer almadığından bu açıklıklar cephe bileşenine dahil edilmiştir. Bir sonraki dönüşüm aşaması olan CityGRIDxml dönüşümü sırasında Çırağan Sarayı için yazılım bu işlemi gerçekleştirememiştir. Örnek veri, firma yetkililerince incelendiğinde hatanın 3B modele ait doku görüntülerine ilişkin dosya adlarında geçen Türkçe karakterlerden kaynaklandığı saptanmıştır ve bu karakterler düzeltilerek giderilmiştir.

Ayrıca 3B modellerde rastlanabilecek problemlerden biri, konum, ölçek ve dönüklük hatalarıdır. Bu hatalar, doğruluğuna güvenilen bir veri seti esas alınarak ilgili yazılımlar ile düzeltilmelidir.

Son olarak, 3B modellerin bir kısmında bütünüyle, bir kısmında ise cephe, taban, çatı gibi bileşenlerinde doku giydirme problemi ortaya çıkmıştır. Bu sorun, modellerin dokulandırılmasında kullanılan kaplama yönteminin CityGRID yazılımında tanımlanan kaplama yöntemleriyle örtüşmemesinden kaynaklanmaktadır. Yazılımda daha çok tercih edilen kaplama yöntemleri tanımlanmıştır. Bu sorun da yazılımda 3B modellere ait görüntü dosyasında yer alan doku görüntülerinin ve yazılımın sunduğu dokuların manuel olarak 3B modele giydirilmesiyle kısmen giderilmiştir.

SONUÇ VE ÖNERİLER

Günümüzde 3B kent modellerinin yaygın kabul görmüş ve birlikte çalışabilirliği destekleyen standartlara sahip olması gerekmektedir. Bu kapsamda CityGML standardı, 3B kent modellerinin temsili ve paylaşımı için OGC tarafından geliştirilmiştir. CityGML ile kentlerin gerçeğe yakın olarak 3B modelleri ve çeşitli uygulama gereksinimlerine yönelik olarak farklı ayrıntı düzeylerinin oluşturulması, bu modellerin mekansal görselleştirmenin ötesinde semantik olarak yapılandırılarak daha ileri mekansal analizlere olanak tanınması ve açık standartlar kullanılarak web ortamında paylaşılabilmesi olanaklı hale gelmiştir.

Bu tez çalışmasında, CityGML standardı incelenmekte ve farklı ayrıntı düzeylerinde bina ve bina parçalarının semantik ve geometrik olarak modellenmesi hedeflenmiştir. Bu standardı bütünüyle destekleyen bir yazılım bulunmamaktadır. Bununla birlikte CityGML'e uyumlu 3B kent modelleri üretmede başarılı sayılabilecek bir kaç yazılım vardır. Bunlardan 3B kent modelleme ve görselleştirme yazılımı CityGRID™, kent nesnelerinin (bina, köprü, kapalı otopark, bodrum) farklı ayrıntı düzeylerinde (LoD0, LoD1, LoD2 ve LoD3) semantik, geometrik ve (dış) görünüş modellemesine olanak tanımaktadır.

Farklı kaynaklardan elde edilen ve CityGML mantığı dikkate alınmadan modellenmiş 3B bina verilerinin, CityGML standardına dönüştürülebilirliğini incelemek amacıyla farklı türde dört adet (müze/cami, okul, kütüphane ve saray) örnek 3B binanın, CityGRID yazılımıyla LoD2 ve LoD3 ayrıntı düzeylerinde semantik, geometrik ve (dış) görünüş modellemesi gerçekleştirilmiştir. Ayrıca bu iki farklı ayrıntı düzeyindeki 3B model,

CityGML standardına dönüştürülmüş ve bu standardı destekleyen başka bir yazılımla ayrıntı düzeyleri görüntülenmiştir.

CityGRID’de geometrik gruplandırma işlemi başta olmak üzere doku giydirme ve CityGML formatına dönüşüm işlemleri, 3B modelleri CityGML standardına dönüştürmede karşılaşılabilecek problemleri ortaya çıkarmıştır. Geometrik gruplandırma işleminde, mevcut modeller oluşturulurken ya da dönüşüm aşamasında bina ve bina parçasına ait bileşenler, küçük parçacıklar halinde olduğundan bu parçaların ait oldukları nesneleri oluşturmak oldukça zaman alıcı ve karmaşıktır. Ayrıca, 3B bina modelleri için çalışmanın ilk aşamasında oluşan doku giydirme problemi, manuel olarak kısmen çözülmüştür.

Bu nedenle CityGML standardına uygun olarak 3B kent modelleri üretebilmek için, 3B bina verilerinin toplanması ve oluşturulması aşamalarında standarda uygunluğun göz önünde bulundurulması gerekmektedir.

Ayrıca TUCBS için INSPIRE spesifikasyonlarını uygulama çabası içindeki ülkemizde 3B bina modellerini oluşturma çalışmalarında, CityGML’in yanı sıra bu standardın kısmen değiştirilmiş ve geliştirilmiş versiyonu olan INSPIRE 3B Bina standardını da dikkate almak mantıklı ve yararlı olacaktır

Gelecek çalışmaların, iç mekan modellerini kapsayan LoD4’ün oluşturulması ve daha kapsamlı ve gelişkin uygulamalara olanak tanımak amacıyla LoD3 ile entegrasyonu ve ayrıca özellikle ülkemiz açısından CityGML standardının farklı tematik uygulamalara (örn. afet yönetimi, kirlilik ve gürültü yayılımı, ısı adaları, uçuş rota planlaması vb.) yönelik eklentilerin geliştirilmesine odaklanacağı düşünülmektedir. Ayrıca, diğer 3B kent nesneleriyle ilgili uygulamalı çalışma sayısı da oldukça azdır.

KAYNAKLAR

- [1] Stadler, A. ve Kolbe, T.K., (2007). "Spatio-Semantic Coherence in The Integration of 3D City Models", Proceedings of the 5th International Symposium on Spatial Data Quality ISSDQ, June 2007, Enschede.
- [2] Mao, B. ve Ban, Y., (2011). "Online Visualization of 3D City Model Using CityGML and X3DOM", Cartographica, 46 (2): 109-114.
- [3] Büyüksalih, İ., Işıkdag U. ve Zlatanova, S., (2013). "Exploring the Processes of Generating LoD (0-2) CityGML Models in Greater Municipality of İstanbul", ISPRS 8th 3D GeoInfo Conference, 27-29 November 2013, İstanbul
- [4] Van den Brink L., Stoter J. ve Zlatanova S., (2013). "UML-Based Approach to Developing a CityGML Application Domain Extension", Transactions in GIS, 17 (6): 920-942
- [5] Kolbe, T.H., (2009). "Representing and Exchanging 3D City Models with CityGML", Lee, J. ve Zlatanova, S. (eds.) 3D Geo-Information Sciences, Lecture Notes in Geoinformation and Cartography, Springer, Berlin.
- [6] Mao, B., Ban, Y. ve Harrie, L., (2011). "A Multiple Representation Data Structure for Dynamic Visualisation of Generalised 3D City Models", ISPRS Journal of Photogrammetry and Remote Sensing, 66: 198-208.
- [7] Gröger G. ve Plümer L., (2012). "CityGML – Interoperable Semantic 3D City Models", ISPRS Journal of Photogrammetry and Remote Sensing, 71: 12-33.
- [8] Kolbe, T.H. ve Gröger, G., (2004). "Towards Unified 3D City Models", J. Schiewe, M. Hahn, M. Madden ve M. Sester (ed.) Challenges in Geospatial Analysis, Integration and Visualization, II. Proc. of Joint ISPRS Workshop, Stuttgart.
- [9] OGC 12-019, (2012). City Geography Markup Language (CityGML) Encoding Standard, OGC.
- [10] Kolbe, T.H., Gröger, G. ve Plümer, L., (2005). "CityGML - Interoperable Access to 3D City Models", International Symposium on Geo-information for Disaster Management, 21-23 March, Delft.

- [11] Götzelmann, T., Guercke, R., Brenner, C. ve Sester, M. (2009). "Terrain-Dependent Aggregation of 3D City Models" ISPRS XXXVIII-2/W11, Lund.
- [12] INSPIRE (2012). "D2.8.III.2 Data Specification on Building Guidelines", INSPIRE Infrastructure for Spatial Information in Europe, JRC, Ispra.
- [13] Özdoğan, Ş. ve Başaraner, M., (2013). "CityGML Standardında Ayrıntı Düzeylerinin Modellenmesi", TMMOB Coğrafi Bilgi Sistemleri Kongresi, 11-13 Kasım 2013, Ankara.
- [14] Yücel, M.A., (2009). Farklı Ayrıntı Düzeylerinde Üç Boyutlu Kent Modelleme ve Uygulanabilirliğinin Araştırılması, Doktora Tezi, YTÜ Fen Bilimleri Enstitüsü, İstanbul.
- [15] Şengül, A., (2010). Extracting Semantic Building Models from Aerial Stereo Images and Conversion to CityGML, Master's Thesis, İTÜ Fen Bilimleri Enstitüsü, İstanbul.
- [16] Karaş, İ.R. ve Ekmekçi, D., (2011). "CityGML Veri Modeline Genel Bakış", TMMOB Elektrik-Elektronik ve Bilgisayar Sempozyumu, 11-13 Kasım 2011, Ankara.
- [17] Olsson, E., (2009). Exporting 3D Geoinformation from BagGIS Database to CityGML, Master's thesis, Royal Institute of Technology (KTH), Division of Geoinformatics, Stockholm.
- [18] Abdul-Rahman, A. ve Pilouk, M., (2007). Spatial Data Modelling for 3D GIS, Springer, Berlin.
- [19] Fan, H., (2010). Integration of Time-Dependent Features within 3D City Model, PhD Thesis, Technical University of Munich, Munich.
- [20] Loidold, M., (2008). "Three-Dimensional GIS", K.K. Kemp (ed.) Encyclopedia of Geographic Information Science. Thousand Oaks: SAGE Publications.
- [21] Edvardsson, K.N., (2013). 3D GIS Modeling Using ESRI's CityEngine, Master's Thesis, Masters Program in Geospatial Technologies, Spain.
- [22] Jarvis, A., (2008). Integration of Photogrammetric and LiDAR Data for Accurate Reconstruction and Visualization of Urban Environments, Master's Thesis, University of Calgary, Calgary.
- [23] Cramer, M., Kresse, W., Skaloud, J., Haala, N., Nittel, S. ve Wallgrün J., (2012). "Data Capture", W. Kresse ve D.M. Danko (ed.) Handbook of Geographic Information, Springer, Berlin.
- [24] Bartelme, N., (2012). "Geographic Information", W. Kresse ve D.M. Danko (ed.) Handbook of Geographic Information, Springer, Berlin.
- [25] Vosselman, G., (2008). "Laser Scanning", S. Shekhar ve H. Xiong, (ed.) Encyclopedia of GIS, Springer, New York.
- [26] Lichti, D., (2008). "Photogrammetry", K.K. Kemp (ed.) Encyclopedia of Geographic Information Science. SAGE Publications, Thousand Oaks.

- [27] Kobayashi, Y., (2006). "Photogrammetry and 3D City Modeling", Proceedings of Digital Architecture, September 2006, Seoul Korea.
- [28] Stewart, M., (2008). "Global Positioning System (GPS)", K.K. Kemp (ed.) Encyclopedia of Geographic Information Science. SAGE Publications, Thousand Oaks.
- [29] Mao, B., (2010). Visualisation and Generalisation of 3D City Models, Licentiate Thesis, Royal Institute of Technology (KTH) Department of Urban Planning and Environment, Stockholm.
- [30] Stadtmodell-Berlin, <http://shop.3d-stadtmodell-berlin.de/en/3d-city-model-berlin-productdescription-quality>, 18 Mart 2014.
- [31] 3D-Stadtmodell-Berlin, <http://shop.3d-stadtmodell-berlin.de/en/gallery>, 18 Mart 2014.
- [32] Over, M., Schilling, A., Neubauer, S. ve Zipf, A., (2010). "Generating Web-based 3D City Models from OpenStreetMap: The Current Situation in Germany", Computers, Environment and Urban Systems, 34: 496-507.
- [33] Mekawy, E., (2010). Integrating BIM and GIS for 3D City Modelling, Licentiate Thesis, Royal Institute of Technology (KTH) Department of Urban Planning and Environment, Stockholm.
- [34] Mekawy, E. ve Anders, Ö., (2010). "Semantic Mapping: An Ontology Engineering Method for Integrating Building Models in IFC and CityGML", 3rd ISDE Digital Earth Summit, 12-14 June, Nessebar.
- [35] What is CityGML?, www.citygml.org, 25 Aralık 2013.
- [36] Glander, T., (2013). Multi-Scale Representations of Virtual 3D City Models, Dissertation, Hasso-Plattner Institut, Berlin.
- [37] Dsilva, M.E.G., (2009). A Feasibility Study on CityGML for Cadastral Purposes, Master's thesis, Eindhoven University of Technology, Department of Mathematics and Computer Science, Eindhoven.
- [38] Fan H. ve Meng L., (2012). "A Three-Step Approach of Simplifying 3D Buildings Modeled by CityGML", International Journal of Geographical Information Science 26(6): 1-18.
- [39] Becker T., Nagel, C. ve Kolbe, T.H., (2011). Integrated 3D Modeling of Multi-utility Networks and Their Interdependencies for Critical Infrastructure Analysis, Kolbe, T.H., König, G. ve Nagel, C. (ed.), Advances in 3D Geoinformation Sciences, Lecture Notes in Geoinformation and Cartography, Springer, Berlin.
- [40] Uden, M. ve Zipf, A., (2013). Open Building Models: Towards a Platform for Crowdsourcing Virtual 3D Cities, Poulit, J., Daniel, S., Hubert, F. ve Zamyadi, A. (eds.) Progress and New Trends in 3D Geoinformation Sciences, Lecture Notes in Geoinformation and Cartography, Berlin.

- [41] Mekawy, M., Östman, A. ve Shahzad, K., (2011). "Towards Interoperating CityGML and IFC Building Models: A Unified Model Based Approach", T.H. Kolbe, G. König, ve C. Nagel (ed.), Advances in 3D Geoinformation Sciences, Lecture Notes in Geoinformation and Cartography, Springer, Berlin.
- [42] UVM Systems Urban Visualisation and Management, Products, <http://www.uvmsystems.com/index.php/en/en-products-overview>, 31 Ocak 2014.
- [43] UVM Systems Urban Visualisation and Management, <http://www.uvmsystems.com>, 2 Şubat 2014.
- [44] UVM Systems Urban Visualisation and Management, 3D City Modelling, <http://www.uvmsystems.com>, 2 Şubat 2014.
- [45] UVM Systems, (2013). Manual Basics, Vienna.
- [46] CityGML, Free Tools for Visualizing and Editing CityGML Files <http://www.citygmlwiki.org/index.php/Freeware>, 7 Şubat 2014.

UML NOTASYONU

CityGML UML diyagramlarının okunabilirliğini artırmak amacıyla, sınıflar farklı UML paketlerine aitse farklı renklerde gösterilmektedir. Renklendirme şeması aşağıda belirtildiği gibi uygulanmaktadır:

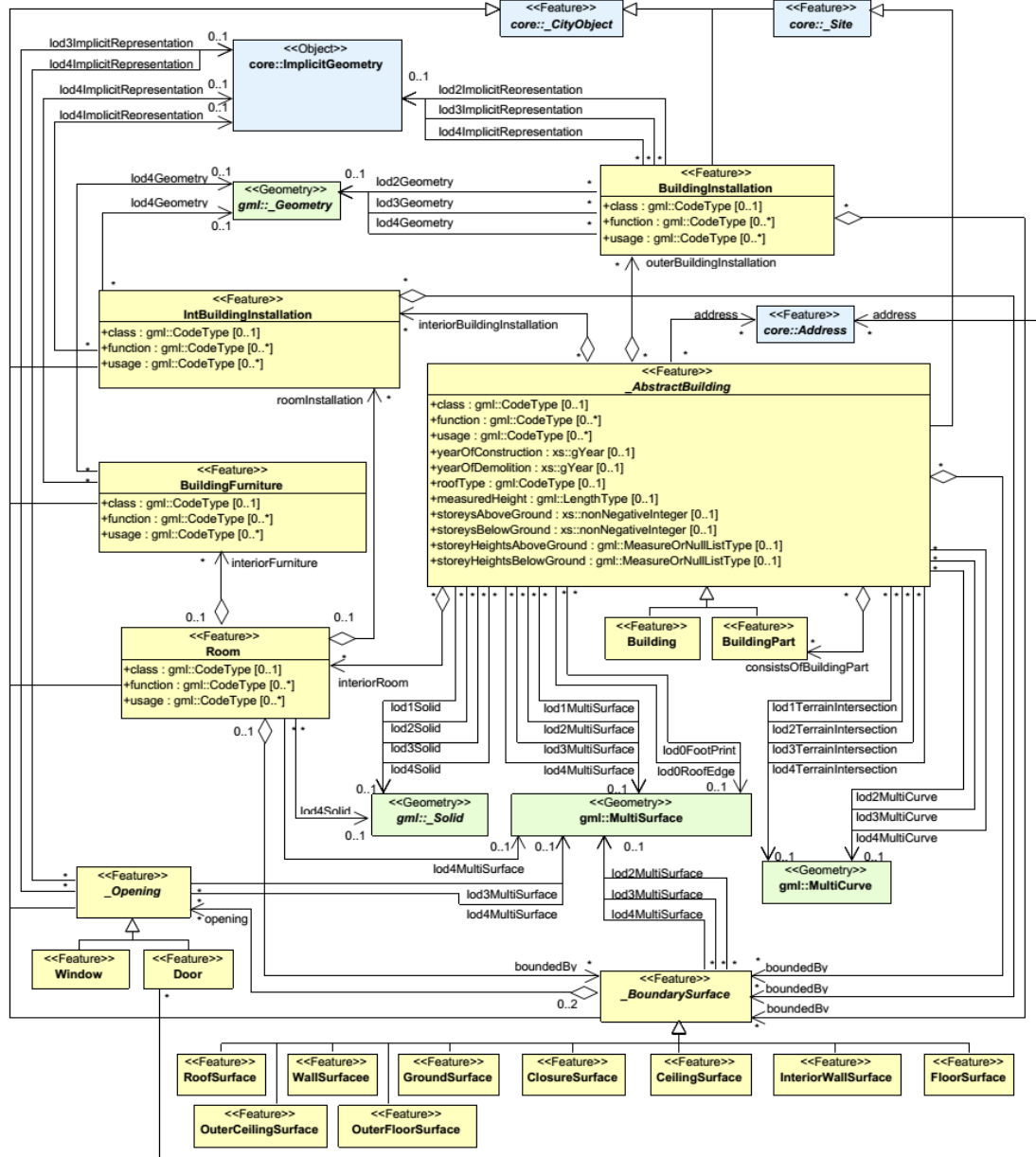
Sarı renkli sınıflar, CityGML Ana Modül UML paketinde tanımlanan sınıfları belirtmek için kullanılmaktadır. Ayrıca, UML diyagramlarında gösterilen sarı sınıflar, Bina modülü ile ilişkilidir.

Mavi renkli sınıflar, sarı renkli sınıflar ile ilişkilendirilmiş olandan farklı CityGML UML paketine aittir. Bu gibi sınıfların UML paketini açıkça göstermek amacı ile onların sınıf adları, bir CityGML modülü ile tekil (eşsiz) olarak ilişkili ad alanı öneki taşırlar. Örnek olarak, *Bina Modülü* konusunda CityGML Ana Modül'den sınıflar mavi renge boyanmıştır ve onların sınıf adlarının başına *core* ön eki eklenmiştir.

Yeşil renkli sınıflar, GML 3'de tanımlanmaktadır ve onların sınıf adlarının başına *gml* ön eki eklenmektedir. Ek-B'de örnek UML diyagramı, UML notasyonunu ve renklendirme şemasını göstermektedir. Bu örnekte sarı sınıflar, CityGML *Bina Modülü* ile ilişkilendirilmiştir, mavi sınıflar CityGML *Core modül*'den gelmektedir. Yeşil sınıflar GML 3 tarafından tanımlanan geometrik öğeleri göstermektedir.

CityGML BİNA MODELİNİN UML DİYAGRAMI

Önekler model öğeleri ile ilişkili XML ad alanlarını belirtmek için kullanılmaktadır. Öneksiz öge adları, CityGML Bina modülü içinde tanımlanır. (Renklerin anlamları için bkz EK-A).



ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Şüheda ÖZDOĞAN
Doğum Tarihi ve Yeri : 20.01.1987 K.MARAŞ
Yabancı Dili : İngilizce
E-posta : suhedaozdogan@gmail.com

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Y. Lisans	Uzaktan Algılama ve CBS	Yıldız Teknik Üniversitesi	-
Lisans	İşletme	Anadolu Üniversitesi	2013
Lisans	Harita Mühendisliği	Yıldız Teknik Üniversitesi	2010
Lise	Fen Bilimleri	K.Maraş Atatürk Lisesi	2004

İŞ TECRÜBESİ

Yıl	Firma/Kurum	Görevi
2013	UVM Systems	3D City Modeller (3B Kent Modelleyici)

YAYINLARI

Bildiri

1. Özdoğan, Ş. ve Başaraner M., (2013). "CityGML Standardında Ayrıntı Düzeylerinin Modellenmesi", 4. Coğrafi Bilgi Sistemleri Kongresi, 11-13 Kasım 2013, Ankara.
2. Özdoğan, Ş. ve Gümüştay Ü., (2011). "Davutpaşa Kampüsü Tarihi Eserlerinin Üç Boyutlu Modellenmesi ve CBS Ortamında Sorgulama", 13. Türkiye Harita Bilimsel ve Teknik Kurultayı, 18-22 Nisan 2011, Ankara.

ÖDÜLLERİ

1. 4. Coğrafi Bilgi Sistemleri Kongresi- En iyi poster ödülü, 11-13 Kasım, Ankara.