Model for Urban spaces A Case Study of Coffee Park-Montreal

Yasaman Niroumand
40056985

Project and Report
ENGR 6971

Project submitted for evaluation to:

Dr. Mazdak Nikbakht

Contents

Abstract

smart cities use information technologies to provide sustainable development in economic, social and environmental urban areas. Over the past years virtual 3D city model have been used for visualization of physical urban objects with respect to their geometrical and graphical properties, however new application domains like urban planning and facility management is a key role for implementing smart city concept and its application. CityGML is the international standard of the Open Geospatial Consortium (OGC) which provide extensible model of the geometrical, semantic, topology, appearance and visual aspects of 3D city objects. CityGML is XML-based format and it defines different standard levels of detail. The aim of this project is to design 3D model of Coffee park located in Montreal, Quebec. The 3D model of case study represents vegetation, city furniture, terrain model, rod and buildings. 3DCityDB-Importer-Exporter and Feature Manipulation Engine **(**FME) tools are used for validation and visualization of GML file respectively.

1 Introduction

City is referred to a dynamic and complex socio-technical system which is approachable to large urban areas such as metropolitan areas. The concept of smart city is defined as “the systematic application to and pervasive penetration of cities by the Information and Communication Technologies (ICTs)”[1] smart cities use information technologies to provide sustainable development in economic, social and environmental urban areas as well as providing an innovative environment for their citizen to have better quality of life. Smart cities are also capable to introduce more efficient and intelligent solution for urban problems such as globalization and sustainability challenges[2]. Representing city objects with respect to their physical characteristics and and geospatial data with a purpose of urban planning and facility management is a key role for implementing smart city concept and its application. 3D models are commonly used for visualization and representation of smart city data and smart cities models must consist of set of data from land, 3D building models, public transportation, urban infrastructure and environment. These models contain spatial information of city assets which will be more expressive when they integrate with urban environment data such as heat, energy, planning zones, routes, and transportation[3].

1.1 Application of 3D City Modeling

Virtual three-dimensional model of city represents buildings, tunnels, bridges, water bodies, city furniture, transportation objects, vegetation with wide range of application like urban planning, environmental and training simulations, navigation, disaster management, energy assessment. According to Biljecki et al.2015 the application of 3D city models can be divided into Visualization-Based Use and Non-visualization Use Cases.

Table 1: Non Visualization-Based use cases of 3D city models[4]

Non-visualization Use case	**Example of an Application**
Estimation of the solar irradiation	Determining the suitability of a roof surface for installing photovoltaic panels
Energy demand estimation	Assessing the return of a building energy retrofit
Aiding positioning	Map matching
Determination of the floorspace	Valuation of buildings
Classifying building types	Semantic enrichment of data sets

Table 2: Visualization-Based use cases of 3D city models [4]

Visualization-Based Use case	Example of an Application
Geo-visualization and visualization enhancement	Flight simulation
Visibility analysis	Finding the optimal location to place a surveillance camera
Estimation of shadows cast by urban features	Determination of solar envelopes
Estimation of the propagation of noise in an urban environment	Traffic planning
3D cadastre	Property registration
Visualization for navigation	Navigation
Urban planning	Designing green areas
Visualization for communication of urban information to citizenry	Virtual tours
Reconstruction of sunlight direction	Object recognition
Understanding SAR images	Interpretation of radar data
Facility management	Managing utilities
Automatic scaffold assembly	Civil engineering
Emergency response	Planning evacuation
Lighting simulations	Planning lighting of landmarks
Radio-wave propagation	Optimizing radio infrastructure
Computational fluid dynamics	Predicting air quality
Estimating the population in an area	Crisis management
Routing	Understanding accessibility

Forecasting seismic damage	Insurance
Flooding	Mitigating damage to utility management
Change detection	Urban inventory
Volumetric density studies	Urban studies
Forest management	Predicting tree growth
Archaeology	Visualizing ancient sites

1.2 Techniques and Methods for 3D City Modeling

Virtual 3D city models also define with different terms like “Cybertown”, “Cybercity”, “Virtual City”, or “Digital City” represent digital model of Earth’ surface and its related object. There are various geomatic techniques for 3D city model generation. These techniques can be categorized as following[5]:

1. Based on Automation
   - Automatic
   - Semi-automatic
   - Manual

2. Based on Data input techniques

   2.1. Photogrammetry based methods

   - Aerial Photogrammetry based model
   - Satellite Photogrammetry based model
   - Close Range Photogrammetry based model

   2.2. LASER scanning based model

   - Aerial Laser based model
   - Terrestrial laser-based model

3. HYBRID METHODS

   Which is the combination of mentioned techniques

2 Problem statement and objectives

Nowadays with development and introduction standardized of data models such as CityGML, many municipalities are creating 3D city models for various applications. For example, New York city provides 3D model of the city comprises all NYC buildings, land parcels, roads, parks, the digital terrain model, and water bodies – all with 3D geometries. The model of New York city is accessible as open data in CityGML, KML, COLLADA, glTF format and 3D web-based visualizations of the entire city mode is provided. [6]

The semantic 3D models of Montreal City are available through open data portal[1]in GML format. These files include buildings in LOD2 with texture files of the Sud-Ouest and Ville-Marie regions. Coffee Park is a local park located 7330 Coffee St, Montreal, Quebec. Interviewing with local residences revealed variety of dissatisfaction with current situation of this park. It was decided to build a webclient for visualization Allowing the public to submit comments about the features of the park. In order to develop this application, the first step is creating 3D model of Coffee park. This project mainly focused on the implementation of CityGML standard to develop 3D model of Coffee Park including buildings, terrain model, railway, roads, city furniture and vegetations. Importer/Exporter tool is used to validate the final GML file.

[1] http://donnees.ville.montreal.qc.ca/dataset/maquette-numerique-batiments-citygml-lod2-avec-textures

3 Previous work on 3D modeling and CityGML

Aditya et al.2017 implemented 3D city modeling using CityGML standard for urban land monitoring of urban tourist areas application. The area of study is Malioboro Street located in Yogyakarta City, Indonesia which is infamous for touristic attraction because of historical and colonial-style buildings located in this street. They develop LOD1 model for buildings through employing different software including 3D City Database, FME, 3DCity DB, Importer/Exporter and their research procedure is summarized in figure They collect georeferenced data from OpenStreetMap (OSM) platform. OSM data are obtained through overpass API. OSM data including 3D building tags were edited by using Kendzi 3D which is JOSM plugin. Collected data were converted from OSM format to CityGML format using OSM2CityGML. This tool saves OSM data in XML format and then convert it to OBJ files with OSM2World and finally FME convert the OBJ format into CityGML format. CityGML data were stored in 3DCityDB database schema and finally for visualization of a webclient was created in Cessium JS.[3]

Preka et al.2016 generated a 3D model in LOD2 of a block of 14 residential buildings at the municipality of Kaisariani, Athen. The aim of their study was to provide a real-life application and solution for land management and registry in the suburb of Kaisariani, Athens. They implemented their model using three software: SketchUp, PostgreSQL/PostGIS and 3D City Database (3DCityDB). The collected data for buildings included address, use, roof type, height and number of storeys and the textures were added to exterior surfaces of the buildings by importing background image from Google Earth to SketchUp software. A CityGML file and its texture files were exported SketchUp through the free plug-in, CityGML Editor 18. They also provide the 3D design of the buildings in .dxf format in order to obtain geometry information. They created an empty database and the schema of the model in PostgreSQL/PostGIS in the format of 3DCityDB and they also define Coordinate Reference system of the case study in Database. The schema of the database in 3D City Database is called public and it has 45 tables which cover city objects with different level of details the empty data base was filled automatically however the data types within the 3DCityDB package was added fine-tuned manually via exported CityGML file. 3DCityDB-Importer-Exporter which is the library of 3DCityDB package were used as a tool to connect 3DCityDB and the PostgreSQL database. The topological and geometrical information of exported CityGML file was added automatically in database tables through the aforementioned key tool. Finally, the database CityGML file and the KML/COLLADA files (visualization files of modeled buildings in Google Earth) were exported by 3DCityDB-Importer-Exporter.in addition a pop-up balloon with all information of the buildings was considered as an extra tool for visualization.[7]

4 Methodology

4.1 Case Study

The case study for this research is Coffee Park located in 7330 Coffee St, Montreal, Quebec. This park has a small playground with play facilities and equipment for children and there is a railway crossing near this park. Another feature of this park is a bike line in coffee Street along the park.

Figure 1: Satellite images of the Coffee Park based on Google Map

4.2 Data and Software

All the data including latitude and longitude of objects are approximately collected from google map. All data points were converted to UTM (Universal Transverse Mercator) coordinate system using online website[2]. The height of buildings was estimated by the number of floors of each building. For this study this software was used: 3D City Database (3DCityDB), 3DCityDB-Importer-Exporter, Feature Manipulation Engine (FME) and sublime text editor.

- **3D City Database (3DCityDB)**

3D City Database(3DCityDB) is a free open geo database to store, represent, and manage virtual 3D city models. The 3D city data support the commercial spatially enhanced relational database management system (SRDBMS) Oracle and PostGIS and it is capable to manage large models in multiple level of details. Data access from user applications to the database can be achieved by using Importer/Exporter tool or by accessing directly the database tables. CityGML datasets can be imported to 3DCityDB and 3D city model can be developed by adding information to the database table in user applications such as Feature Manipulation Engine from Safe Software (FME)[8].

- **Importer/Exporter tool**

Importer/Exporter tool allows importing, exporting and validation of CityGML dataset. It is possible to export KML, COLLADA, and glTF formats of 3D models by using Import/Export tool in order to visualize 3D city and landscape models in geoinformation systems (GIS) or digital virtual globes like Google Earth or CesiumJS Virtual Globe. Importer/Exporter comes with export Spreadsheet Generator Plugin (SPSHG) which can be used which allows to export thematic data of 3D objects into tables in CSV and Microsoft Excel format and these files can be uploaded as online spreadsheets within Google Cloud[8].

Feature Manipulation Engine (FME)

FME (Feature Manipulation Engine) is a powerful transformation engine produced by Safe Software. Data can be converted into several formats including CityGML by using FME. FME includes key transformers for CityGML such as Attribute Creator and Geometry Property Setter.[9]

---

[2] https://www.latlong.net/lat-long-utm.html

4.3 CityGML Overview

Over the past years virtual 3D city model have been used for visualization of physical urban objects with respect to their geometrical and graphical properties, however new application domains like environmental and training simulations, urban planning and facility management, disaster management and homeland security, and personal navigation require more comprehensive semantical, topological and thematical information as well as semantic interrelationships of 3D object[10].

In order to represent and exchange the 3D city models, a standard 3D data model is required. CityGML is the international standard of the Open Geospatial Consortium (OGC) and it provide extensible model of the geometrical, semantic, topology, appearance and visual aspects of 3D city objects such as buildings, roads, rivers, bridges, vegetation, terrain and city furniture. In addition, CityGML can represent the relation between those features[11].

"CityGML is an open data model and XML-based format for the storage and exchange of virtual 3D city models. It is an application schema for the Geography Markup Language version 3.1.1 (GML3), the extendible international standard for spatial data exchange issued by the Open Geospatial Consortium (OGC) and the ISO TC211. The aim of the development of CityGML is to reach a common definition of the basic entities, attributes, and relations of a 3D city model. This is especially important with respect to the cost-effective sustainable maintenance of 3D city models, allowing the reuse of the same data in different application fields"[12].

CityGML is an application schema of GML which is based on markup language XML for interoperable representation of geometry and topology of features. CityGML works perfectly with spatial data infrastructure and it facilitates exchanging data via geo-web services[11].

In CityGML relevant entities of the urban space can be documented with respect to their semantics, 3D geometry, 3D topology, and appearances in five level of details (LOD 0-4) (figure 4). CityGML provides UML class diagrams to explain classes and attributes of physical objects which can be exchanged by using XML schema[13].

CityGML employs UML class diagram to define classes and attributes of object and it also provides XML schema for the file exchange format. Classes in CityGML are categorized in 12 semantic modules (figure 3). The core module is a fundamental module which is referenced by other modules in many areas such as Building, Bridge, Transportation, CityObjectGroup, Appearance, Generic, CityFurniture, Relief, Vegetation, Tunnel, LandUse, and WaterBody[11].

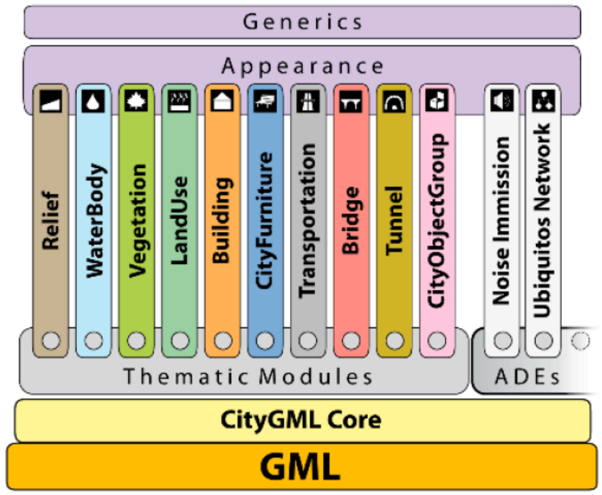

Figure 2: The CityGML Thematic Modules [3]

Various modulus which can be documented in CityGML are as following[14]:

- Appearance: textures and materials
- Bridge: bridge-related structures, possibly split into parts
- Building: the exterior and possibly the interior of buildings with individual surfaces that represent doors, windows, etc.
- CityFurniture: benches, traffic lights, signs, etc.
- CityObjectGroup: groups of objects
- Generics: other types that are not explicitly covered
- LandUse: areas that reflect different land uses, such as urban, agricultural, etc.
- Relief: the shape of the terrain
- Transportation: roads, railways and squares
- Tunnel: tunnels, possibly split into parts
- Vegetation: areas with vegetation or individual trees
- WaterBody: lakes, rivers, canals, et

Thematic module	Feature types	Feature types (cont'd.)
Core	_CityObject (LoD0–4)	
Relief	ReliefFeature (LoD0–4)	TINRelief (LoD0–4)
	MassPointRelief (Lo 0–4)	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)

Figure 3: Thematic modules and associated feature types including the LoDs in which they are defined[11].

4.3.1 Level of Detail in CityGML

CityGML is also capable to implement geometry, topology, and semantics properties of 3D objects in five level of detail (LOD0-4). Level of details demonstrate the complexity of information and data related to defined 3D object (figure 5). Representation and visualization of 3D objects in different level of details depend on two main components:

1. Data collection method
2. Application of city 3D model

The level of data quality plays a critical role in defining level of detail[15].

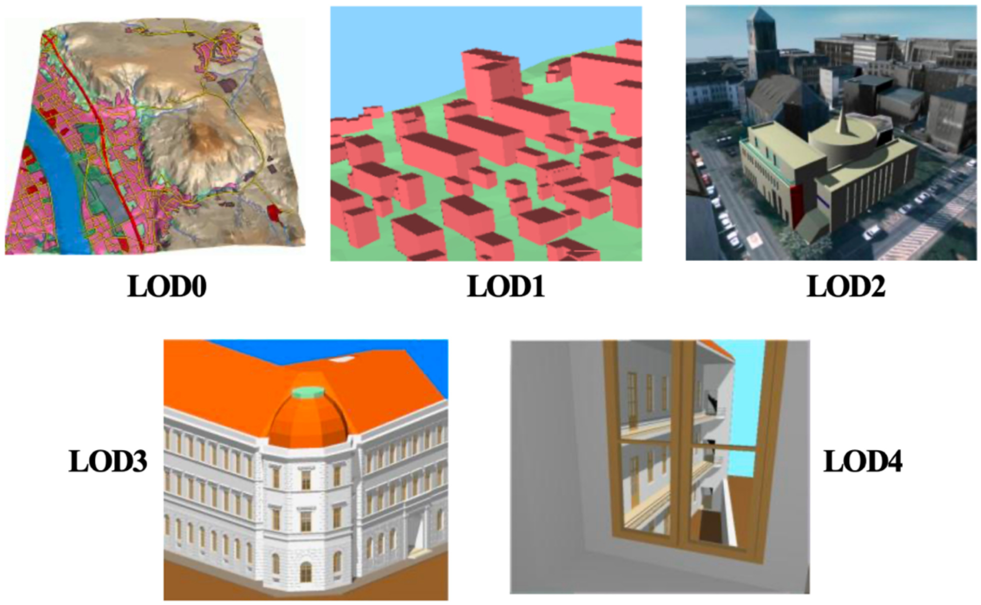

Figure 4: The five levels of detail (LOD) defined by CityGML[12]

For example, level of detail for a building can be defined as follow[12, 14, 15] (figure 6):

“LOD 0: Footprint with its elevation or roof edge polygons which does not have volumetric features

LOD 1: Blocks model comprising prismatic buildings with flat roofs derived by extruding

a footprint to a uniform height

LOD 2: has differentiated roof structures and thematically differentiated boundary surfaces.

LOD 3: architectural models with detailed wall and roof overhang structures potentially including openings, doors and windows and façade details

LOD 4: LOD3 models representing interior features like rooms, doors, stairs and furniture.” [12, 14, 15]

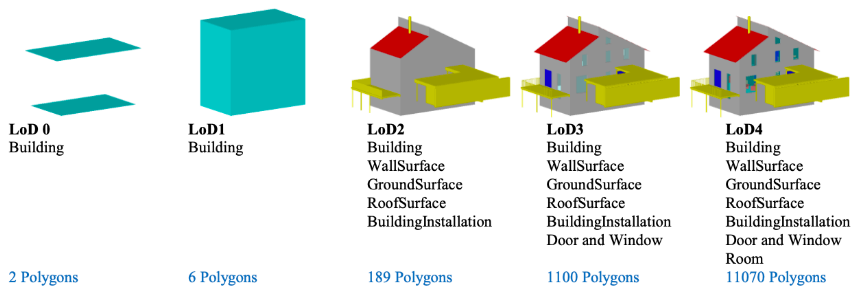

Figure 5: Geometric and semantic Level of Detail for buildings[15]

On the other hand, LOD is distinguished by the accuracy and dimension of features and accuracy is defined as standard deviation σ of the absolute 3D point coordinates.

	LOD0	**LOD1**	**LOD2**	**LOD3**	**LOD4**
Model scale description	regional, landscape	city, region	city, city districts, projects	city districts, architectural models (exteri-or), landmark	architectural models (interi-or), landmark
Class of accuracy	lowest	low	middle	high	very high
Absolute 3D point accuracy (position / height)	lower than LOD1	5/5m	2/2m	0.5/0.5m	0.2/0.2m
Generalisation	maximal generalisation	object blocks as generalised features; > 6*6m/3m	objects as generalised features; > 4*4m/2m	object as real features; > 2*2m/1m	constructive elements and openings are represented
Building installations	no	no	yes	representative exterior features	real object form
Roof structure/representation	yes	flat	differentiated roof structures	real object form	real object form
Roof overhanging parts	yes	no	yes, if known	yes	yes
CityFurniture	no	important objects	prototypes, gener-alized objects	real object form	real object form
SolitaryVegetationObject	no	important objects	prototypes, higher 6m	prototypes, higher 2m	prototypes, real object form
PlantCover	no	>50*50m	>5*5m	< LOD2	<LOD2
…to be continued for the other feature themes					

Figure 6: LOD 0-4 of CityGML with their proposed accuracy requirements[15]

There are different techniques of data acquisition to generate level of details. LOD one models for building can be generated from extrusion of its footprint and the height of the buildings can be achieved from attributes available in volunteered geoinformation and cadastre or simply can be predicted through the number of building's stories. Higher level of details also can be constructed in CityGML using OpenStreetMap (OSM) and point-cloud data[3].

4.4 Modeling procedure

The semantic 3D models of Montreal City are available through open data portal[3]in GML format. The first attempt in this project was to find the GML file of Coffee park in existing files in order to complete the model and add missing objects such as vegetation and city furniture. A new empty database was created in PostgreSQL/PostGIS. In order to store and manage spatial features, it is necessary to create ***PostGIS*** Extension with the new database. The 3D City Database comes with a set of stored procedures known as the CITYDB package which one of them is citydb_pkg schema which in this project I faced difficulty to find this schema as a pre-installed package in pgAmin4. To tackle this problem the mentioned schema was created manually, and the instruction is provided in readme file which can be find here[4]. Please be noted that this project was ran on MacBook Pro 2019 device and all the instructions are in respect of MacOS Catalina version 10.15.1.

3DCityDB Importer/Exporter tool makes connection and communication between 3DCityDB and the PostgreSQL database. After implementing the connection between 3DCityDB and the PostgreSQL database the existing GML files of Montreal City were imported to 3DCityDB Importer/Exporter. "The Importer/Exporter GUI offers a 2D map window that allows the user to display the overall bounding box calculated from the city model content stored in each 3D City Database". The map is provided by *OpenStreetMap* (OSM) service. Through bounding box is it possible to search the address or geo location (geographic latitude/longitude coordinates). After searching through all the existing file and directories I realized that the GML file of case study does not exist, so I aimed to create the 3D model of Coffee park based on CityGML standard (explained in sections 3.4.1-3.4.4) manually from scratch. However on of the difficulties during this project was lack of detailed information and data of objects that I aimed to model. The only resource to gather the need data was Google Map service.

[3] http://donnees.ville.montreal.qc.ca/dataset/maquette-numerique-batiments-citygml-lod2-avec-textures
[4] https://github.com/Yasaman-Niroumand/Coffee-park

4.4.1 Digital Terrain Model (DTM)

The Digital Terrain Model (DTM) is provided by the thematic extension module Relief and class ReliefFeature in CityGML. The terrain is defined as a TIN (Triangulated Irregular Network, TINReflief) and implemented by gml: TriangulatedSurface class.[12]

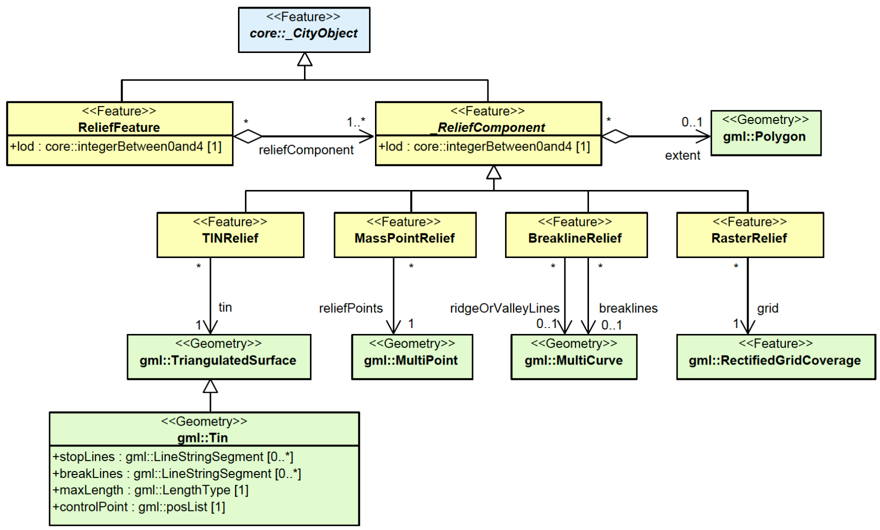

Figure 7: UML diagram of the Digital Terrain Model in CityGML. Prefixes are used to indicate XML namespaces associated with model elements. Element names without a prefix are defined within the CityGML Relief module[12].

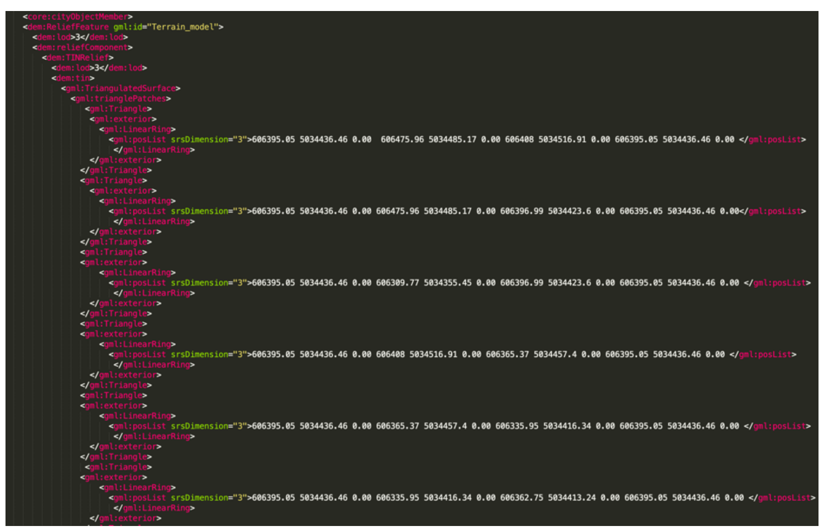

Figure 8: GML code used for Terrain mode

4.4.2 Building model

Building as a most detailed thematic model of CityGML is provided by the thematic extension module building. _Abstract Building is the main class of this model and it is subclass of the root class _CityObject. The following attributes are provided for the buildings which are the attributes of building class: class of the building, the function, the roof type and the measured height and these attributes are of the class building. the block of the buildings. Geometric properties of the buildings are defined with gml: MultiSurface. In addition, building's exterior blocks are represented by the class _BoundarySurface with its function including WallSurface, GroundSurface and RoofSurface.[12]

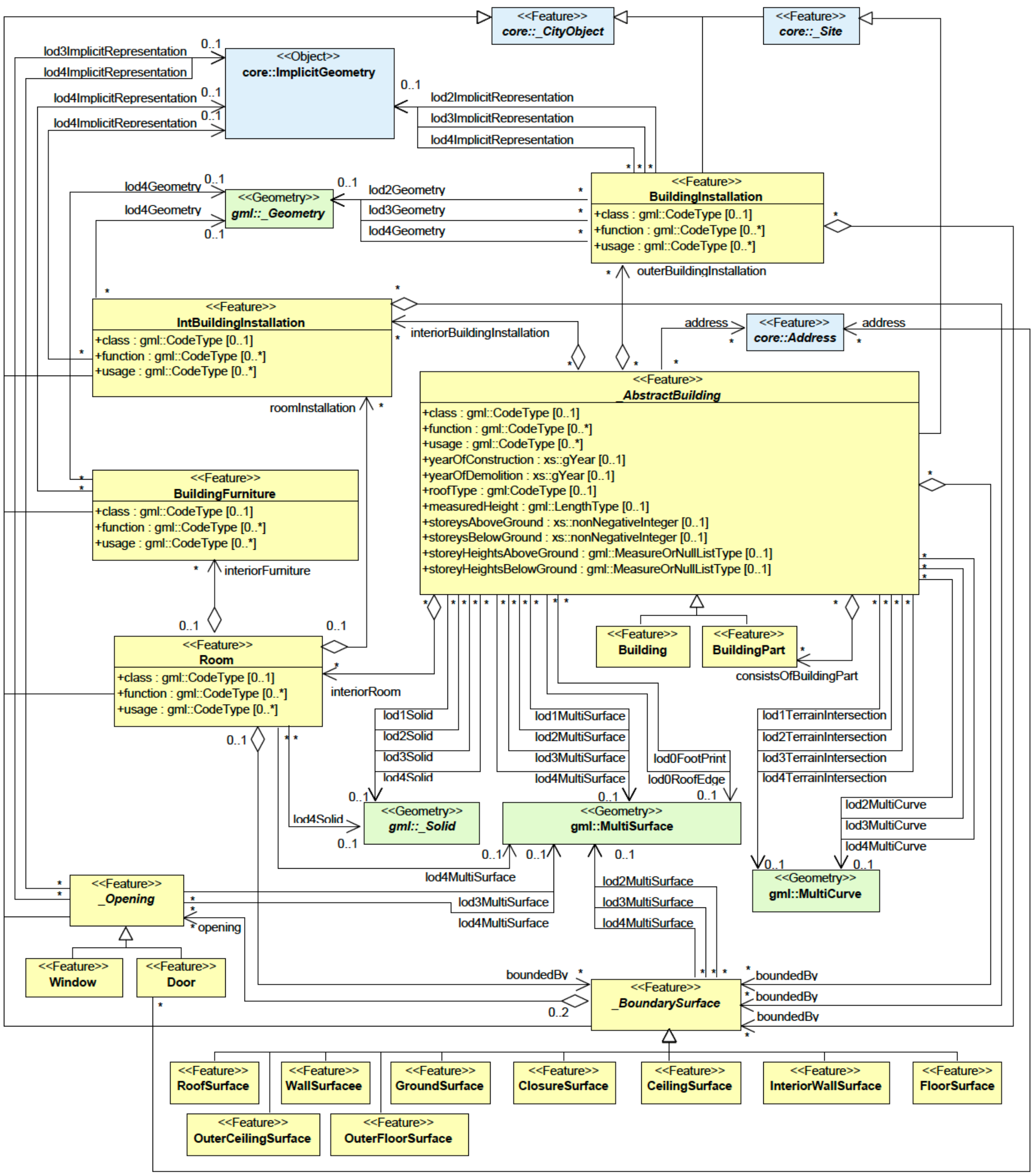

Figure 9: UML diagram of CityGML's building model. Prefixes are used to indicate XML namespaces associated with model elements. Element names without a prefix are defined within the CityGML Building module[12].

```xml
<core:cityObjectMember>
  <bldg:Building gml:id="Building01">
    <bldg:roofType>1000</bldg:roofType>
      <bldg:function>1000</bldg:function>
    <bldg:measuredHeight>8.6</bldg:measuredHeight>
    <bldg:boundedBy>
      <bldg:GroundSurface gml:id="GroundSurface_01_01">
        <bldg:lod2MultiSurface>
          <gml:MultiSurface>
            <gml:surfaceMember>
              <gml:Polygon>
                <gml:exterior>
                  <gml:LinearRing>
                    <gml:posList srsDimension="3">606369.09 5034491.91 0.00 606363.53 5034482.82 0.00 606349.48 5034490.36 0.00 606354.56 5034499.56 0.0 606369.09
                    5034491.91 0.00  </gml:posList>
                  </gml:LinearRing>
                </gml:exterior>
              </gml:Polygon>
            </gml:surfaceMember>
          </gml:MultiSurface>
        </bldg:lod2MultiSurface>
      </bldg:GroundSurface>
    </bldg:boundedBy>
      <bldg:boundedBy>
    <bldg:WallSurface gml:id="WallSurface_01_01">
      <bldg:lod2MultiSurface>
          <gml:MultiSurface>
            <gml:surfaceMember>
              <gml:Polygon>
                <gml:exterior>
                  <gml:LinearRing>
                    <gml:posList srsDimension="3">606369.09 5034491.91 0.00 606369.09 5034491.91 8.6 606354.56 5034499.56 8.6 606354.56 5034499.56 8.6 606369.09 5034491.91
                    0.00</gml:posList>
                  </gml:LinearRing>
                </gml:exterior>
              </gml:Polygon>
            </gml:surfaceMember>
          </gml:MultiSurface>
        </bldg:lod2MultiSurface>
      </bldg:WallSurface>
    </bldg:boundedBy>
<bldg:boundedBy>
   <bldg:WallSurface gml:id="WallSurface_01_02">
      <bldg:lod2MultiSurface>
          <gml:MultiSurface>
```

Figure 10: GML code used for building models

4.4.3 Vegetation objects

The vegetation model of CityGML is defined by the thematic extension module Vegetation and class SolitaryVegetationObject. Two attributes including function and species attributes are provided for single trees and the geometric properties are defined by ImplicitGeometry.[12]

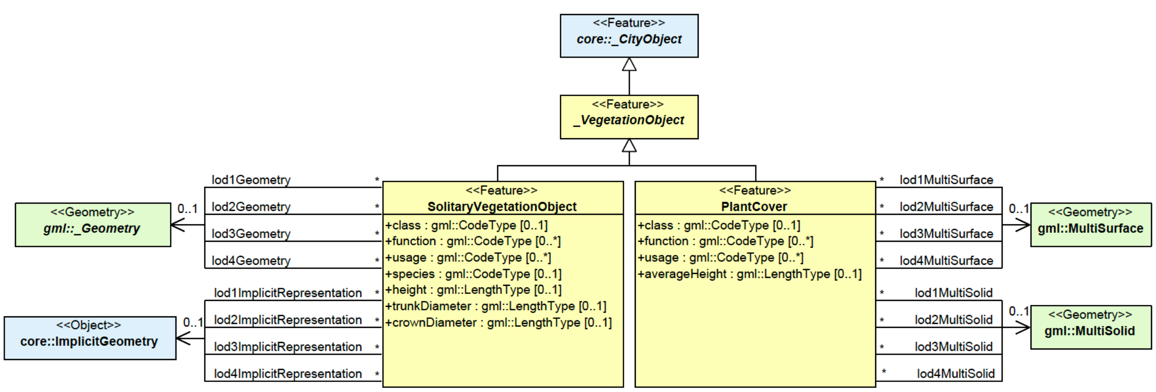

Figure 11: UML diagram of vegetation objects in CityGML. Prefixes are used to indicate XML namespaces associated with model elements.[12]

```xml
  <core:cityObjectMember>
<veg:SolitaryVegetationObject gml:id="vegetation_01">
<veg:function>1060</veg:function>
<veg:species>9999</veg:species>
<veg:lod1ImplicitRepresentation>
<core:ImplicitGeometry>
<core:referencePoint>
<gml:Point>
<gml:pos srsDimension="3">606404.73 5034447.18 0.00 </gml:pos>
</gml:Point>
</core:referencePoint>
</core:ImplicitGeometry>
</veg:lod1ImplicitRepresentation>
</veg:SolitaryVegetationObject>
</core:cityObjectMember>
```

Figure 12: GML code used for Vegetation models

3.5.4. City furniture

The city furniture object like traffic signs, benches, streetlamps are modeled for case study. The city furniture model of CityGML is defined by the thematic extension module CityFurniture and class CityFurniture which includes attributes such as function and class.[12]

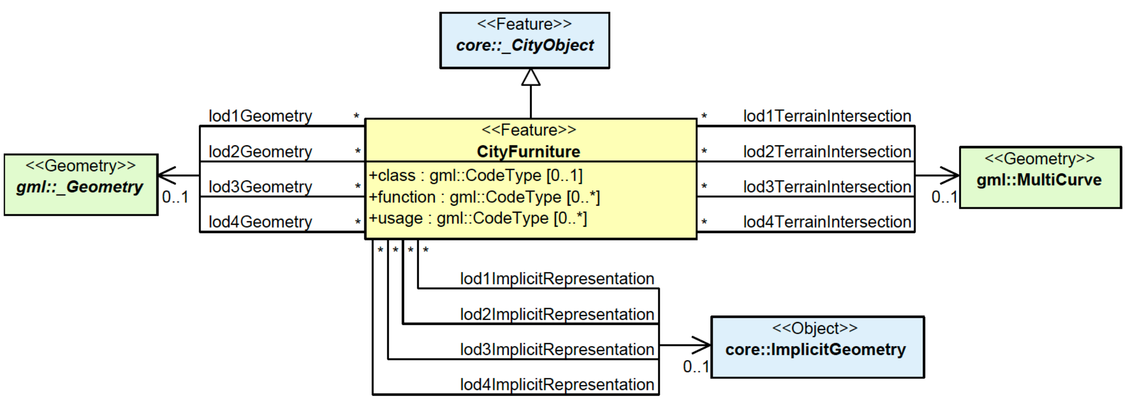

Figure 13UML diagram of city furniture objects in CityGML[12].

```xml
<core:cityObjectMember>
<frn:CityFurniture gml:id="Road_sign_01">
<frn:function>1070</frn:function>
<class>1000</class>
<frn:lod1ImplicitRepresentation>
<core:ImplicitGeometry>
<core:referencePoint>
<gml:Point>
<gml:pos srsDimension="3">606414.09 5034508.57 0.00</gml:pos>
</gml:Point>
</core:referencePoint>
</core:ImplicitGeometry>
</frn:lod1ImplicitRepresentation>
<frn:lod2ImplicitRepresentation>
</frn:lod2ImplicitRepresentation>
</frn:CityFurniture>
</core:cityObjectMember>
```

Figure 14: GML code used for Cityfurniture models

```xml
  <core:cityObjectMember>
    <tran:Road gml:id="Street01">
      <gml:name>Rue West Broadway</gml:name>
      <tran:lod2MultiSurface>
        <gml:MultiSurface  srsDimension="3">
          <gml:surfaceMember>
            <gml:Polygon >
              <gml:exterior>
                <gml:LinearRing>
                  <gml:posList>606501.49 5034497.16 0.00 606487.66 5034486.70 0.00 606485.10 5034485.77 0.00 606482.14 5034485.16 0.00 606479.09 5034484.89 0.00 606475.96
                  5034485.17 0.00 606408.00 5034516.91 0.00 606405.56 5034518.20 0.00  606316.06 5034566.03 0.00 606317.35 5034568.38 0.00 606477.64 5034487.64 0.00
                  606486.84 5034489.46 0.00 606491.39 5034492.76 0.00 606501.49 5034497.16 0.00 </gml:posList>
                </gml:LinearRing>
              </gml:exterior>
            </gml:Polygon>
          </gml:surfaceMember>
        </gml:MultiSurface>
      </tran:lod2MultiSurface>
</tran:Road>
  </core:cityObjectMember>
```

Figure 15: GML code used for Road models

5 Results

The 3D model of Coffee Park was developed mainly for visualization. CityGML standard was used to support the representation of geometry and semantic attributes of 3D objects. The current GML file contains terrain (LOD 0), buildings (LOD 2), roads and railway (LOD0), vegetation (LOD 0) and city furniture (LOD 0) such as lights and traffic signs. Exterior parts of the buildings of the case study area were modeled as blocks in LOD2 that represent the wall, roof and ground surfaces of their exterior. Collected data for these buildings consist of height, address and their types.

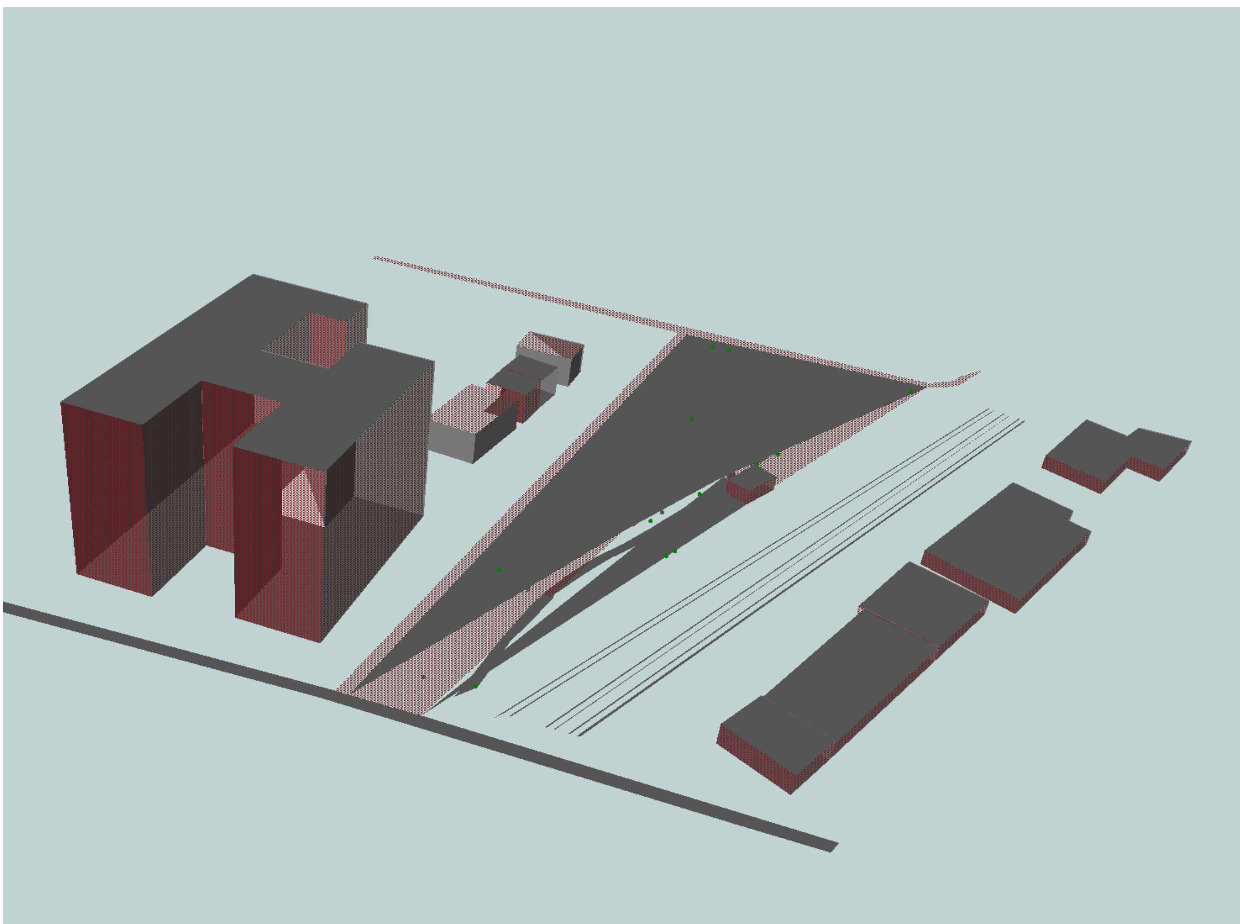

Figure 16 3D model of Coffee Park exported from FME

6 Conclusion

The demand for semantic 3D city models has increased due to numerous smart city application such as urban planning, decision making, environmental and training simulations, navigation, disaster management, energy assessment. Typically, 3D city models represent urban objects with respect to their graphical and geometrical properties for visualization purposes. City Geography Markup Language (CityGML) standard has been developed to represent the thematic properties, taxonomies and aggregations as well as graphical appearance of city objects. The available 3D model of Montréal city represents buildings in LOD 2 predominantly.

This project implements the application of CityGML standard for 3D modeling of Coffee park located in Montreal city. We focused on expanding the domain of CityGML application in 3D modeling to provide semantic model of different city objects such as terrain model, vegetation, city furniture and roads besides the buildings. the result of this project is a CityGML format file which can be visualized in FME. For future work CityGML file can be stored in 3DcityDB. 3DcityDB as a free source consists of database schema and CityGML data can be imported as tables in database and it has capability to export other 3D format for visualization purposes. Furthermore, a web client can be provided for more visualization and smart city applications.

References

[1] M. Finger and M. Razaghi, "Conceptualizing Smart Cities," *Informatik-Spektrum,* vol. 40, no. ARTICLE, pp. 6-13, 2017.

[2] M. Basiri, A. Z. Azim, and M. Farrokhi, "Smart city solution for sustainable urban development," *European Journal of Sustainable Development,* vol. 6, no. 1, pp. 71-71, 2017.

[3] T. Aditya and D. Laksono, "LOD 1: 3D CityModel for Implementing SmartCity Concept," in *Proceedings of the 2017 International Conference on Information Technology*, 2017, pp. 136-141.

[4] F. Biljecki, J. Stoter, H. Ledoux, S. Zlatanova, and A. Çöltekin, "Applications of 3D city models: State of the art review," *ISPRS International Journal of Geo-Information,* vol. 4, no. 4, pp. 2842-2889, 2015.

[5] S. P. Singh, K. Jain, and V. R. Mandla, "Virtual 3D city modeling: techniques and applications," *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* no. 2, pp. 73-91, 2013.

[6] "3D City Model of New York City." https://www.lrg.tum.de/gis/projekte/new-york-city-3d/#c838 (accessed.

[7] D. Preka and A. Doulamis, "3D BUILDING MODELING IN LOD2 USING THE CITYGML STANDARD," *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences,* vol. 42, 2016.

[8] T. H. Kolbe, C. Nagel, and J. Herreruela, "3D City Database for CityGML," *Addendum to the 3D City Database Documentation Version,* vol. 2, no. 1, 2019.

[9] E. Dimopoulou, E. Tsiliakou, V. Kosti, G. Floros, and T. Labropoulos, "Investigating integration possibilities between 3D modeling techniques," in *Proceedings of the 9th 3DGeoInfo Conference*, 2014, pp. 1-16.

[10] T. H. Kolbe, "Representing and exchanging 3D city models with CityGML," in *3D geo-information sciences*: Springer, 2009, pp. 15-31.

[11] G. Gröger and L. Plümer, "CityGML–Interoperable semantic 3D city models," *ISPRS Journal of Photogrammetry and Remote Sensing,* vol. 71, pp. 12-33, 2012.

[12] G. Gröger, T. H. Kolbe, C. Nagel, and K.-H. Häfele, "OGC city geography markup language (CityGML) encoding standard," 2012.

[13] Z. Yao *et al.*, "3DCityDB-a 3D geodatabase solution for the management, analysis, and visualization of semantic 3D city models based on CityGML," *Open Geospatial Data, Software and Standards,* vol. 3, no. 1, pp. 1-26, 2018.

[14] K. A. Ohori, F. Biljecki, K. Kumar, H. Ledoux, and J. Stoter, "Modeling cities and landscapes in 3D with CityGML," in *Building Information Modeling*: Springer, 2018, pp. 199-215.

[15] J. Benner, A. Geiger, G. Gröger, K.-H. Häfele, and M.-O. Löwner, "Enhanced LOD concepts for virtual 3D city models," in *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences. Proceedings of the ISPRS 8th 3D GeoInfo conference & WG II/2 workshop*, 2013, pp. 51-61.

[16] G. Gröger, T. H. Kolbe, and A. Czerwinski, "Candidate OpenGIS CityGML Implementation Specification (City Geography Markup Language)," *Open Geospatial Consortium Inc, OGC,* 2007.

Appendix

Code list of the _AbstractBuilding attribute roofType	
http://www.sig3d.org/codelists/standard/building/2.0/_AbstractBuilding_roofType.xml	
1000	flat roof
Code list of the _AbstractBuilding attributes function and usage	
http://www.sig3d.org/codelists/standard/building/2.0/_AbstractBuilding_usage.xml http://www.sig3d.org/codelists/standard/building/2.0/_AbstractBuilding_function.xml	
1000	residential building
1050	residential- and business building
Code list of the TransportationComplex attributes function and usage	
http://www.sig3d.org/codelists/standard/transportation/2.0/TransportationComplex_function.xml http://www.sig3d.org/codelists/standard/transportation/2.0/TransportationComplex_usage.xml	
1000	road
1240	bikeway/cycle-path
Code list of the CityFurniture attributes function and usage	
http://www.sig3d.org/codelists/standard/cityfurniture/2.0/CityFurniture_function.xml http://www.sig3d.org/codelists/standard/cityfurniture/2.0/CityFurniture_usage.xml	

1200	rubbish bin
1070	road sign
1150	streetlamp, latern or candelabra
Code list of the CityFurniture attribute class	
http://www.sig3d.org/codelists/standard/cityfurniture/2.0/CityFurniture_class.xml	
1000	traffic
1030	other

Modeling and GML features

1.Geomery

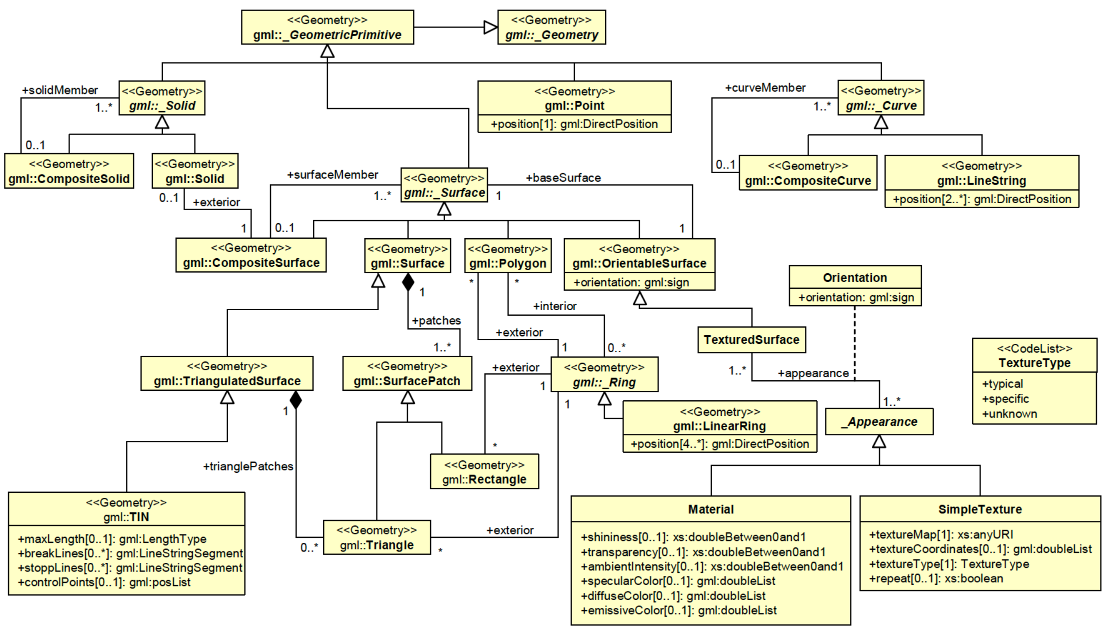

Figure 17: UML diagram of CityGML's geometry model (subset and profile of GML3): Primitives and Composites[12]

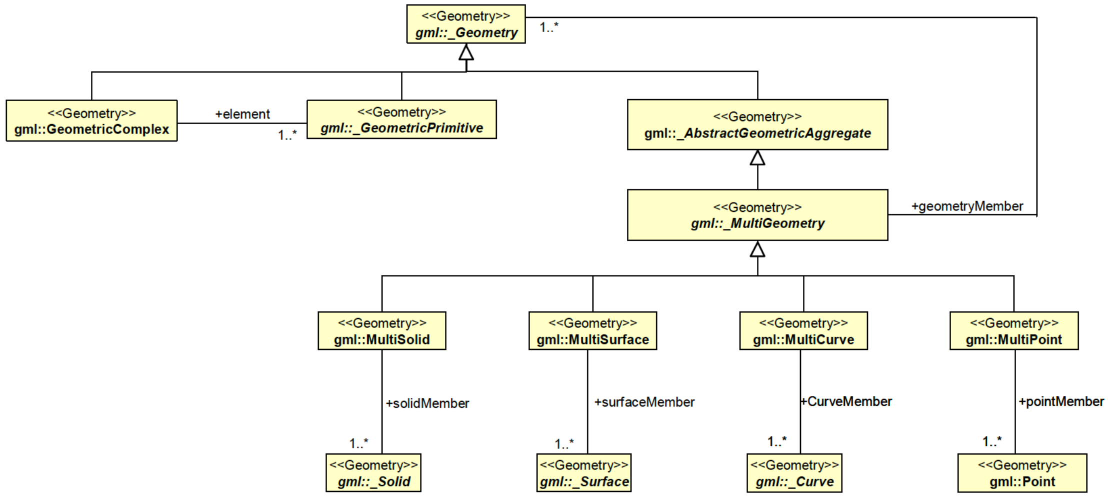

Figure 18: UML diagram of CityGML's geometry model: Complexes and Aggregates[12]

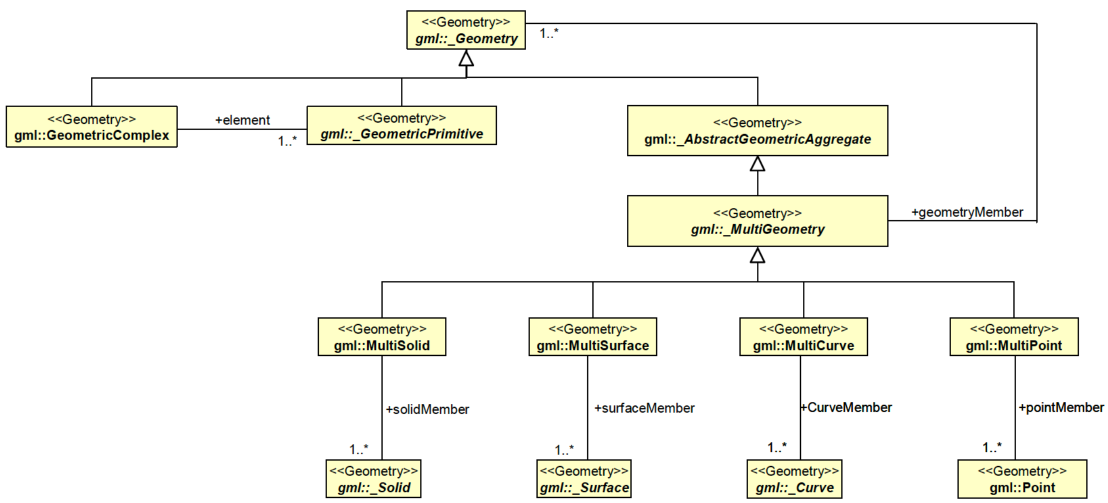

Figure 19: UML diagram of ImplicitGeometries [16].