



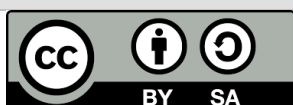
L1.3 Estándares de datos 3D

Apuntes

Autor(es)/Organización(es):

Ariana Kubart, Ocellus Information Systems AB, Sweden

Licencia



<https://creativecommons.org/licenses/by-sa/4.0/>

Versión

Version 2.0

Date: Mayo 2025

Resumen

La última lección de este bloque describe los modelos 3D de ciudades a nivel de datos. Introduce al estudiante en el modelo conceptual CityGML y describe sus módulos y cómo se pueden utilizar en diversos aspectos del modelado de ciudades. Profundiza en ciertas representaciones, por ejemplo, geométricas, topológicas o temporales, porque el conocimiento de estas es importante para comprender el proceso de integración BIM-GIS. Además, la conferencia proporciona información sobre diferentes codificaciones CityGML, así como otros formatos 3D.

Resultados de aprendizaje

Al final de esta clase, se espera que el alumno sea capaz de:

- Nombra varias formas en que se pueden almacenar los datos 3D, con codificaciones CityGML enfocadas
- Resumir las partes principales del modelo conceptual CityGML y cómo se pueden utilizar
- Comprender los aspectos de CityGML que son importantes para la conversión hacia y desde BIM



Competencias esperadas al ingresar a la clase magistral

Conocimientos intermedios de SIG

- L1.1 Conceptos de modelado 3D
- L1.2 Modelos semánticos de ciudad

Carga de trabajo esperada

26 diapositivas con contenido de aprendizaje, aproximadamente 5 horas

Financiado por la Unión Europea. Las opiniones y puntos de vista expresados solo comprometen a su(s) autor(es) y no reflejan necesariamente los de la Unión Europea o los de la Agencia Ejecutiva Europea de Educación y Cultura (EACEA). Ni la Unión Europea ni la EACEA pueden ser considerados responsables de ellos.

Contenido

Modelos semánticos de ciudades en 3D.....	4
CityGLM Estándar	5
Modelo Conceptual 6 de CityGLM	6
Módulos de clase en el modelo conceptual.....	7
Módulos de aspecto específico en CM.....	8
Información semántica en los modelos 3D	9
Esquema de agregación	10
Modelización semántico-geométrica coherente	11
Niveles de detalle, LOD, en CityGML.....	12
Niveles de detalle, LoD II	13
Versiones anteriores de CityGML, 1.0 y 2.0	14
Representación geométrica	15
Representación de topología en CityGML 3.0.....	16
Espacios físicos y lógicos	17
Exteriores, Interiores y Jerarquías.....	18
Coordenadas y elevación.....	19
4D – dimensión temporal.....	20
Extensiones de dominio de aplicación, ADE en CityGML.....	21
Calidad de los datos.....	22
Codificaciones de CityGML.....	23
Codificación XML	24
Codificación CityJSON I.....	25
Codificación CityJSON II.....	26
Codificación de base de datos de ciudades 3D	27
Otros formatos 3D.....	28
PaísInfra.....	29
Inspirar edificios 3D.....	30
Referencias.....	31



Modelos semánticos de ciudades en 3D

- Proporcionar información geográfica, topográfica y semántica sobre los objetos
- Interacciones entre objetos
- Descomposición jerárquica en partes más pequeñas (por ejemplo, edificio, pared, ventana)
- Puede ser complejo y cubrir grandes áreas



Se puede agregar mucha información semántica a un modelo 3D
Modelo 3D de Nancy, Francia, Google Earth

4

Modelos semánticos de ciudades en 3D

Los modelos semánticos de ciudad proporcionan información geográfica, topográfica y semántica sobre objetos en áreas urbanas (ver Lección 1.2). Los objetos semánticos se pueden representar espacialmente mediante múltiples geometrías y en diferentes LOD. Además, los objetos se pueden descomponer en partes más pequeñas en varios niveles y esta descomposición es jerárquica (por ejemplo, edificio-pared-ventana).

Los modelos 3D listos para usar deben almacenarse en algún lugar y ser accesibles para los usuarios finales. Al mismo tiempo, los modelos semánticos pueden ser muy complejos y abarcar grandes áreas.

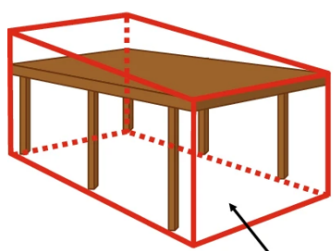
Significa que una geodatabase 3D para un modelo 3D debe ser capaz de hacer frente tanto a la jerarquía semántica como a la geométrica, para gestionar el dataset grande y permitir el acceso, las actualizaciones y las consultas de los datos.



Estándar CityGML

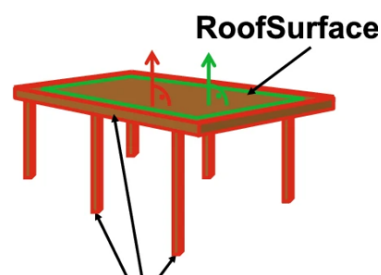
- CityGML es un modelo de datos abiertos del Open Geospatial Consortium (OGC)
- Destinado a representar modelos 3D semánticos
- La versión actual es la 3.0, aprobada en 2021

Carport in LOD1



OccupiedSpace

Carport in LOD2/3



OccupiedSpace

Representation of a carport as *OccupiedSpace* in different LODs. From: [CityGML 3.0: New Functions Open Up New Applications](#)

5

Estándar CityGLM

¿Cómo se codifica todo eso en la computadora?

Existe un modelo de datos abiertos estandarizado por el Open Geospatial Consortium (OGC), destinado a representar los modelos semánticos 3D. El estándar se llama CityGML y la versión actual es la 3.0, aprobada en 2021.

La propia definición de OGC suena: "CityGML es un modelo de información semántica común para la representación de objetos urbanos en 3D que se puede compartir en diferentes aplicaciones".

¿Cómo funciona? Vamos a verlo más de cerca en esta lección.

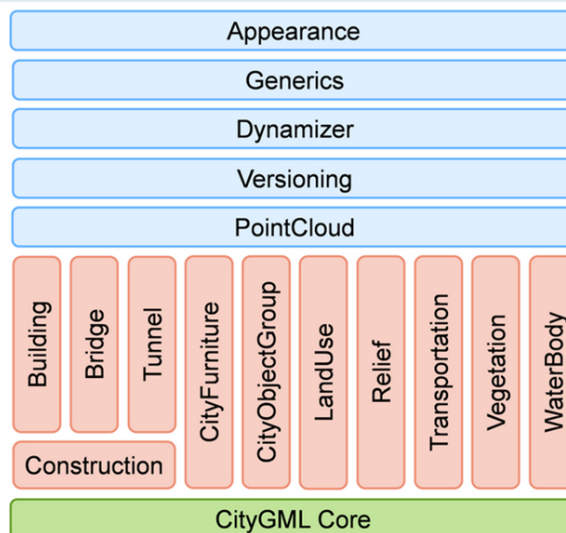
[CityGML - Open Geospatial Consortium \(ogc.org\)](https://ogc.org/)

[CityGML 3.0: New Functions Open Up New Applications | PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science \(springer.com\)](#)

Estándares de datos 3D

Modelo Conceptual CityGML (CM)

- Los objetos comunes de la ciudad deben describirse de la misma manera en todos los modelos
- CityGML define las clases de los objetos más importantes, cómo descomponerlos y los atributos útiles
- Estas clases se definen en el modelo conceptual CityGML



CityGML Conceptual Model, [CityGML - Open Geospatial Consortium \(ogc.org\)](http://www.opengeospatial.org/ogc/citygml)

6

S

Modelo conceptual CityGLM

Muchos objetos, como edificios, calles o puentes, son comunes en todas las ciudades. Por lo tanto, deben describirse de la misma manera en todos los modelos. De lo contrario, por ejemplo, los "edificios" podrían llamarse "casas" en otra ciudad y no habría interoperabilidad.

Por lo tanto, define CityGML sus propias clases de los objetos más importantes, cómo descomponerlos y, a menudo, algunos atributos útiles que se incluirán. Estas clases se definen en el Modelo Conceptual (CM) de CityGML.

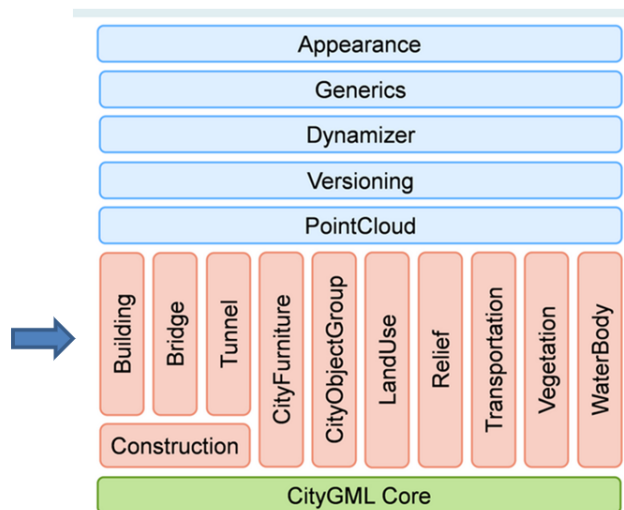
El modelo conceptual se visualiza mediante la figura de la diapositiva.

La parte básica del modelo conceptual es el módulo principal, que se muestra en verde. El módulo básico comprende los conceptos y componentes básicos de una ciudad virtual. Por lo tanto, siempre debe implementarse en cualquier modelo de ciudad en 3D.

Estándares de datos 3D

Módulos de clase en CM

- Módulos de clase de extensión, mostrados verticalmente en rosa
- Los módulos son clases temáticas específicas de los objetos más importantes
- No es necesario incluir todos los módulos de extensión en un modelo 3D



CityGML Conceptual Model, [CityGML - Open Geospatial Consortium \(ogc.org\)](http://CityGML - Open Geospatial Consortium (ogc.org))

7

Módulos de clase en el modelo conceptual

Además del módulo Core, están los módulos de la clase de extensión, que se muestran verticalmente en color rosa.

Cada módulo de extensión cubre una clase temática específica para los objetos más importantes dentro de la ciudad virtual 3D. Estos incluyen construcciones, relieves, por ejemplo, en forma de DTM (modelo digital del terreno), cuerpos de agua, etc., como se muestra en la figura.

Los tres módulos *Edificio*, *Puente* y *Túnel* modelan estructuras civiles y comparten conceptos comunes. Es por ello que se agrupan dentro del *módulo de Construcción*.

El módulo Transporte define clases para la representación de la infraestructura de tráfico. Permite utilizar CityGML para simulaciones de tráfico y conducción, sistemas de asistencia a la conducción, conducción autónoma, así como con sistemas de gestión de instalaciones viales y ferroviarias.

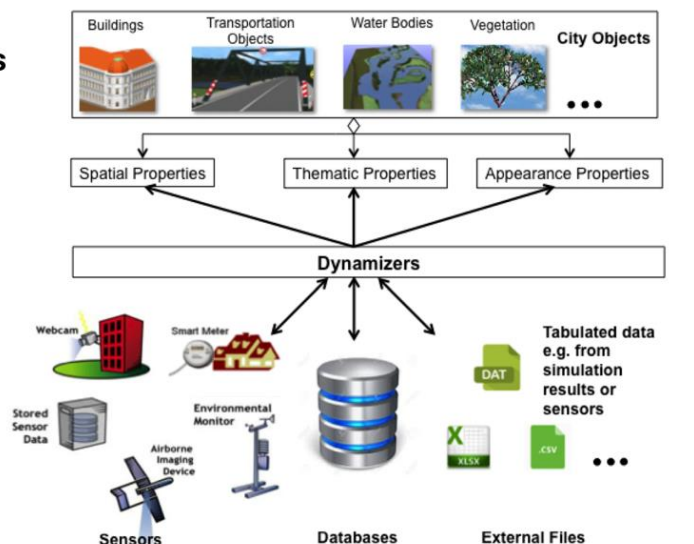
Un modelo de ciudad no necesita soportar todas las clases (=módulos de extensión). Puede emplear solo un subconjunto de acuerdo con las necesidades específicas de los modelos.

Por ejemplo, si una aplicación funciona exclusivamente con datos de construcción, es suficiente solo con los *módulos Core*, *Construction* y *Building*.

Estándares de datos 3D

Módulos de aspectos específicos

- Los cinco módulos verticales de color azul en el CM
- Añade aspectos específicos de modelado, por ejemplo, texturas, colores, tiempo, versiones del modelo...
- Se puede utilizar junto con todos los módulos de extensión



Conceptual representation of *Dynamizers* allowing enhancing the properties of city objects,
From: [CityGML 3.0: New Functions Open Up New Applications](#)

8

Módulos de aspecto específico en CM

Los cinco módulos de color azul añaden aspectos específicos de modelado. Estos que se pueden utilizar junto con todos los módulos de extensión.

El módulo *Apariencia* representa, por ejemplo, texturas y colores de objetos urbanos.

El módulo *PointCloud* proporciona conceptos para representar la geometría de objetos mediante nubes de puntos 3D.

El módulo *Genéricos* define objetos, atributos y relaciones genéricos. Los objetos genéricos representan objetos 3D, que no están cubiertos por las clases modeladas explícitamente. Del mismo modo, los atributos genéricos y las relaciones son esos adicionales.

El control de versiones agrega conceptos para la representación de versiones simultáneas de los objetos.

El módulo *Dynamizer* permite representar las propiedades de los objetos mediante datos de series temporales y vincularlos con sensores o archivos externos.



Información semántica en los modelos 3D

- Todos los objetos pertenecen a una clase o pueden definirse como objetos "genéricos"
- El objeto puede ser representado por semántica (= propiedades no espaciales), geometría 3D, topología 3D, apariencias y cambios a lo largo del tiempo
- FeatureID único y obligatorio para cada objeto



Ejemplo de la Universidad Técnica de Delft de cinco tipos de techos: planos, a dos aguas, a cuatro aguas, piramidales y cobertizos.
GitHub - tudelft3d/3dbook: Libro para el curso GEO1004: Modelado 3D del entorno construido

9

Información semántica en los modelos 3D

En CityGML, todos los objetos de ciudad son modelados por las clases, como hemos visto anteriormente. De este modo, estos objetos "saben" lo que son y dónde están.

Combinando los diversos módulos CityGML, cada objeto puede ser representado por su semántica (= propiedades no espaciales), geometría 3D, topología 3D, apariencias (por ejemplo, características de la superficie) y sus cambios a lo largo del tiempo.

Cada objeto tiene un *featureID* único y obligatorio. También puede tener un *identificador* opcional. El *identificador* es idéntico para todas las versiones del mismo objeto, como una ventana o una tabla. Eso facilita las consultas temáticas, las tareas de análisis o la minería de datos espaciales, mucho más allá de la visualización.

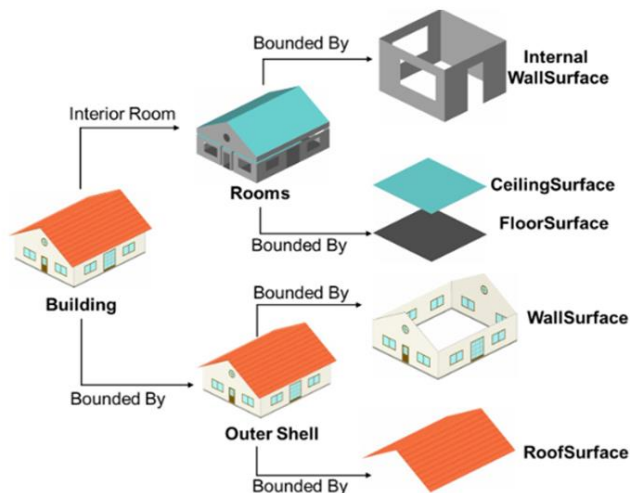
Los atributos para clasificar objetos a menudo se restringen a un conjunto de valores discretos. Por ejemplo, hay un número limitado de tipos de techos. Se especifican como listas de códigos externos. Sin embargo, el usuario puede redefinir estas listas de códigos externas.

Además, los objetos que no están explícitamente cubiertos por las clases del modelo conceptual pueden representarse como objetos genéricos y atributos genéricos.

Estándares de datos 3D

Esquema de agregación

- Puede haber interrelaciones jerárquicas entre los objetos de la ciudad
- Jerarquía de agregación: todo el edificio y su descomposición en partes (ver figura)
- Útil para consultas, simulaciones y análisis



Vista general de un edificio representado en formato CityGML. De Malhotra et al (2021): Herramienta de código abierto para transformar los niveles de detalle de CityGML

Esquema de agregación

Los modelos de ciudad deben reflejar la complejidad de los objetos de la ciudad y sus interrelaciones.

Los objetos se pueden relacionar entre sí mediante diferentes tipos de relaciones en CityGML, y estas relaciones suelen ser jerárquicas.

Los objetos complejos, como los edificios o los objetos de transporte, suelen estar formados por piezas. Estas partes son características individuales en sí mismas, e incluso pueden descomponerse aún más. Ya hemos mencionado en L1.2 el ejemplo de edificio jerárquico – parte del edificio – superficie del techo, superficie de la pared e instalación del edificio – puerta, ventana como subcategoría de la superficie de la pared. Esto se conoce como jerarquías de agregación de componentes, que, en otras palabras, definen las relaciones parte-todo entre las características.

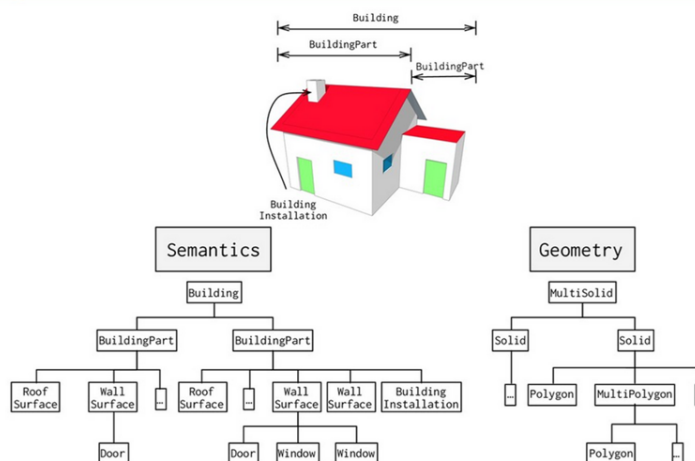
No es solo un objeto que se puede descomponer. Incluso toda el área urbana puede descomponerse semánticamente en, por ejemplo, uso de la tierra, cubierta vegetal, edificios.

Esta estructura jerárquica es necesaria para consultas, simulaciones y análisis.

Semántica coherente -Modelado geométrico

- El edificio y sus partes = jerarquía de agregación semántica
- También existe una jerarquía geométrica: ubicación, forma, extensión...
- Es crucial que la semántica y la geometría de los objetos correspondientes estén vinculadas entre sí

Spatio-semantic coherence



De Ledoux (2018): [CityGML](#) y sus dos codificaciones [CityGML](#) y [CityJSON](#)

11

Modelado semántico-geométrico coherente

Por lo tanto, los objetos de la ciudad están representados por características, como edificios o ventanas. También pueden incluir atributos, relaciones y jerarquías entre las entidades. Todo eso es información semántica.

Sin embargo, a los objetos también se les asigna su ubicación, forma y extensión, es decir, por su geometría.

Por lo tanto, el modelo consta de dos jerarquías: la semántica y la geométrica, como se muestra en la figura. Los objetos correspondientes (y sus partes) están vinculados por relaciones. Es crucial que estos vínculos coincidan y que encajen entre sí.

Es posible navegar en ambas jerarquías y entre ellas arbitrariamente, para responder consultas temáticas y geométricas o para realizar análisis.

Estándares de datos 3D

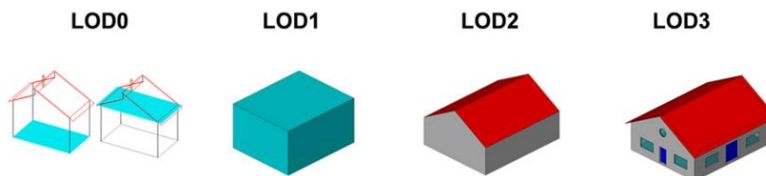
Nivel de detalle, LoD I

LOD0 – Modelo altamente generalizado

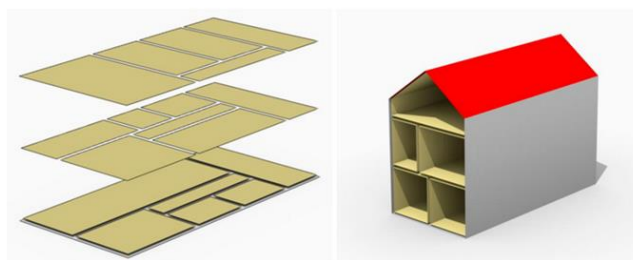
LOD1 – Modelo de bloques / objetos de extrusión

LOD2 – Modelo realista, pero aún generalizado

LOD3 – Modelo muy detallado



Representation of the same real-world building in the Levels of Detail 0-3



Floor-plan representation in LoD0 (left) and in LoD2 (right). From: <https://www.gim-international.com/content/news/citygml-3-0-conceptual-model-approved-as-official-ogc-standard>

17

Niveles de detalle, LOD, en CityGML

Cada objeto puede tener diferentes representaciones espaciales al mismo tiempo. Es proporcionado por cuatro niveles de detalle predefinidos (LOD 0-3).

En resumen, el nivel de detalle es la cantidad de información en el modelo que retrata el mundo real.

Los LOD son:

LOD0 – Modelo altamente generalizado;

LOD1 – Modelo de bloque / objetos de extrusión;

LOD2 – Modelo realista, pero aún generalizado;

LOD3 – Modelo muy detallado.

Por ejemplo, un edificio también puede ser abstraído por una huella o impresión de techo (LOD0), por un sólido 3D con techo plano (LOD1) hasta una visualización detallada en LOD3.

Estándares de datos 3D

Nivel de detalle, LoD II

- LoDs are applicable to both interior and exterior
-
- Individual building or whole neighbourhoods
- possible to combine different LODs in same model

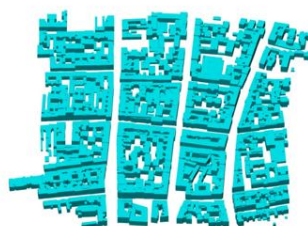
Visual representation of several housing-blocks in Vienna (a) underlying image from Google Maps, (b) representation of open LoD2 CityGML model, (c) LoD1 model transformed from LoD2, (d) LoD0 CityGML models. From Malhotra et al (2021)



(a)



(b)



(c)



(d)

Niveles de detalle, LoD II

Los modelos LoD1 son fáciles de reconstruir: la huella de un edificio se puede extruir a su altura. Esta altura puede ser, por ejemplo, el promedio de todos los puntos LIDAR dentro de la huella.

LoD2 incluye la forma generalizada del techo. Como tales, los modelos LoD2 son útiles para estimar el potencial fotovoltaico. Por lo general, se obtienen con técnicas fotogramétricas y se derivan automáticamente.

LoD3 es un modelo detallado que contiene aberturas (ventanas y puertas), chimeneas y otros detalles de fachada. Los modelos en LoD3 generalmente se obtienen con una conversión a partir de modelos BIM o de escaneo láser terrestre. La presencia de ventanas y otros detalles los hace útiles en aplicaciones como simulaciones energéticas.

Estos LOD son aplicables tanto al interior como al exterior y se pueden combinar. Los planos de planta de los edificios son representaciones LOD0 de los interiores de los edificios.

Es posible combinar diferentes niveles de detalle en un mismo modelo 3D. Por ejemplo, la envolvente exterior de un edificio se puede representar en LOD2 y los elementos interiores, como habitaciones o escaleras, en LOD1. Incluso es posible modelar la envolvente exterior de un edificio en LOD1, mientras se representa la estructura interior en LOD2 o LoD3.

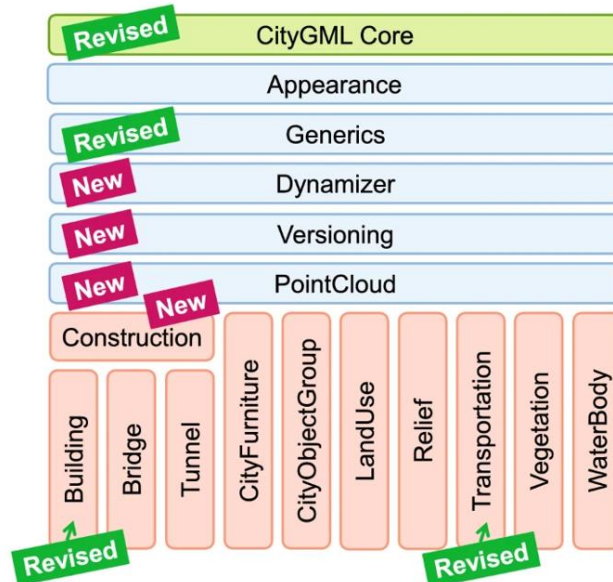


Estándares de datos 3D

Versiones anteriores de CityGML, 1.0 y 2.0

- CityGML 3.0 es una evolución de las versiones 1.0 y 2.0.
- 1.0 y 2.0 todavía se usan en muchos modelos, pero se pueden actualizar a 3.0
- La versión 3.0 tiene muchas capacidades nuevas

Partes nuevas y revisadas de la versión 3.0, en comparación con la 2.0. De: Kutzner et al (2020) [CityGML 3.0: New Functions Open Up New Applications](#) | PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science (springer.com)



14

Versiones anteriores de CityGML, 1.0 y 2.0

CityGML 3.0 es una evolución de las versiones anteriores 1.0 y 2.0.

Tiene muchas capacidades nuevas. Estos incluyen una integración mucho mejor con BIM, la capacidad de representar espacios interiores en diferentes niveles de detalle (LOD), soporte para datos de sensores dinámicos y la capacidad de extender el modelo de información a extensiones de dominio de aplicación, ADE (como discutiremos más adelante).

Como hay muchos modelos creados con CityGML 1.0 y 2.0, estas especificaciones no quedarán obsoletas. Sin embargo, estos datos existentes de CityGML se pueden actualizar a CityGML 3.0. con el fin de alcanzar las nuevas posibilidades.

Estándares de datos 3D

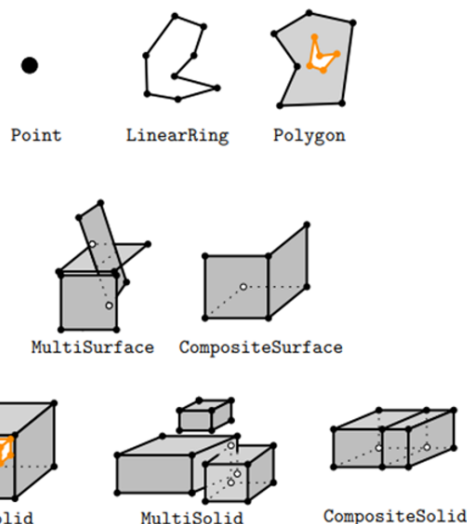
Representación de geometría

Las propiedades espaciales de todos los objetos CityGML están representadas por clases de geometría definidas en la norma ISO 19107

Entre ellas se encuentran:

Geometrías primitivas: puntos, curvas, superficies y sólidos
Diferentes tipos de geometrías agregadas

Todas las geometrías guardadas en el módulo Core



Algunas de las primitivas de CityGML, incluidos agregados y compuestos.
De Qhori et al (2020-2022) [Releases · tudelft3d/3dbook \(github.com\)](https://github.com/tudelft3d/3dbook/releases)

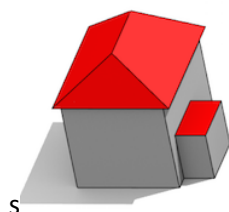
15

Representación de geometría

Las propiedades espaciales de todos los objetos CityGML se representan utilizando las clases de geometría definidas en la norma ISO 19107. Estos incluyen geometrías primitivas, es decir, puntos individuales, curvas, superficies y sólidos.

Además, existen diferentes tipos de geometrías agregadas: agregados espaciales (*MultiPoint*, *MultiCurve*, *MultiSurface*, *MultiSolid*) y compuestos (*CompositeCurve*, *CompositeSurface*, *CompositeSolid*).

Los objetos se definen mediante atributos que definen sus subelementos. Estos subelementos se combinan para formar el objeto completo. Suena complicado, así que vamos a ilustrarlo con una figura:



S

[GitHub - tudelft3d/3dbook](https://github.com/tudelft3d/3dbook): Book for the course GEO1004: 3D modelling of the built environment

La geometría de este edificio está compuesta por la geometría de la casa (*CompositeSolid*) y el garaje/anexo de la derecha (*Sólido*). La geometría de la casa se descompone aún más en la geometría del techo (*Sólido*) y la geometría del cuerpo de la casa (*Sólido*).

En CityGML 3.0 todas las representaciones geométricas se definen solo en el módulo Core.

3D Data Standards

Representación de topología



Esquema [CityGML](#) que muestra la relación entre la ventana y la habitación y el edificio. De: [Salheb \(2019\)](#) Conversión automática de [CityGML](#) a IFC

- La topología sigue la descomposición completa, similar a la geometría
- Relaciones entre elementos bien definidos en [CityGML 3.0](#)
- Espacios: objetos del mundo real
- Límites espaciales: delimite y conecte los espacios (por ejemplo, superficie de la pared, superficie de la carretera...)

16

Representación de topología en CityGML 3.0

El modelo de topología de CityGML3.0 sigue la descomposición completa de topologías n-dimensionales hasta el nivel de nodos, de manera similar a la descomposición geométrica.

El esquema de la relación entre elementos está bien definido. Por ejemplo, la relación entre una ventana y una habitación se define en el esquema que se muestra en la figura.

CityGML 3.0 también mapea todos los objetos de la ciudad en dos conceptos semánticos: estos dos son 'espacio' y 'límite espacial'.

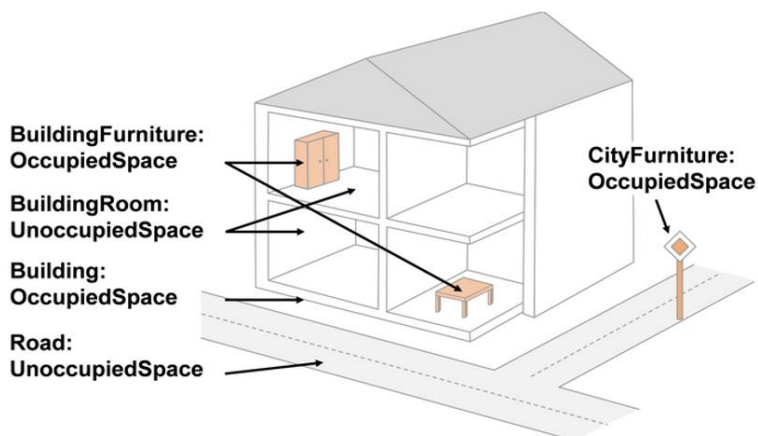
Un *espacio* es un objeto del mundo real con una extensión volumétrica, como edificios, masas de agua, árboles, habitaciones y espacios de tráfico.

Los límites del espacio delimitan y conectan *los espacios*. Algunos ejemplos son las superficies de las paredes y los tejados que limitan un edificio, la superficie del agua como límite entre la masa de agua y el aire, la superficie de la carretera o el modelo digital del terreno.

Esto es importante a la hora de convertir CityGML a IFC (el formato BIM abierto).

Espacios Físicos y Lógicos

- Los espacios físicos están delimitados por objetos físicos, por ejemplo, edificios, árboles...
- Ocupado o desocupado (ver figura)
- Espacios lógicos: temáticos, p. ej., distrito de la ciudad, unidad de construcción



Espacios ocupados y desocupados. De: [Kutzner et al \(2020\) CityGML 3.0: Las nuevas funciones abren nuevas aplicaciones](#)

17

Espacios físicos y lógicos

Los espacios se subdividen a su vez en espacios físicos y espacios lógicos.

Los espacios físicos están total o parcialmente delimitados por objetos físicos. Los edificios y las habitaciones son espacios físicos, ya que están delimitados por muros y losas. Los espacios de tráfico de las carreteras son espacios físicos, ya que están delimitados por las superficies de las carreteras contra el suelo.

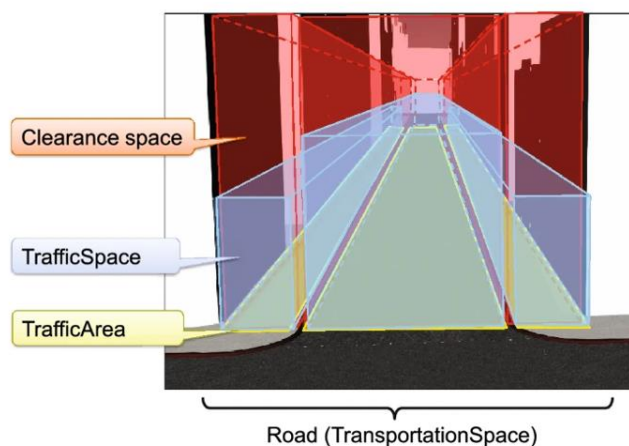
Los espacios lógicos, por el contrario, se definen de acuerdo con consideraciones temáticas, como la unidad de construcción, el distrito de la ciudad y las zonas de seguridad en los aeropuertos.

Los espacios físicos se clasifican a su vez en espacios ocupados y espacios desocupados.

Los espacios ocupados representan objetos volumétricos físicos, por ejemplo, edificios, puentes, árboles, mobiliario urbano y masas de agua. Los espacios desocupados son las habitaciones de los edificios y los espacios de tráfico. Veámoslo en la Figura xx.

Exteriores, Interiores y Jerarquías

- Cada edificio puede tener exterior, interior, subterráneo
- La clasificación y descomposición automáticas solo son posibles para las partes externas (visibles), no para las internas
- Las solicitudes requieren información más detallada
- Posibilidad de clasificación semiautomática para unidades más pequeñas



Descomposición del espacio de transporte; Es difícil conseguirlo de forma totalmente automatizada. De: Kutzner et al (2020) CityGML 3.0: Las nuevas funciones abren nuevas aplicaciones

18

S

Exteriores, Interiores y Jerarquías

Tanto el exterior como el interior de un edificio pueden ser descritos por CityGML.

Además, el interior de un edificio puede tener diferentes habitaciones (BuildingRoom), diferentes pisos o unidades (BuildingStorey y BuildingUnit), pero también instalaciones (por ejemplo, chimeneas, antenas, balcones, etc.).

Se puede modelar incluso bajo tierra, como espacios huecos y capas de roca geológica, utilizando el concepto de espacios y límites espaciales.

Uno de los principales problemas para la aplicación es cómo agregar las jerarquías de objetos, es decir, cómo descomponerlos en las partes básicas y cómo tratar los interiores. Como las nubes de puntos registran partes observables, las descomposiciones de objetos solo son posibles para las superficies visibles. Por ejemplo, los edificios se pueden descomponer en paredes, techos y superficies del suelo de forma más o menos automática. También es relativamente fácil clasificar los otros objetos de la ciudad, como la vegetación o los cuerpos de agua. Estos se pueden convertir en clases GML.

Sin embargo, las diversas aplicaciones, como los gemelos digitales utilizados para simulaciones y análisis, requieren conocimientos semánticos más detallados. En el mejor de los casos, deben ser modelos de ciudades 3D semánticos completos, que contengan información semántica y espacial estructurada jerárquicamente. Desafortunadamente, sigue siendo un desafío fundamental obtenerlos mediante el procesamiento automático de imágenes.

Una solución para áreas más pequeñas o edificios individuales es definir las jerarquías (semi) manualmente.

Estándares de datos 3D

Coordenadas y elevación

Todas las geometrías de [CityGML](#) deben:

utilizar valores de coordenadas 3D

- estar absolutamente georreferenciado
- Pueden incluir modelos de terreno

2.5D: solo una coordenada Z para todos los edificios



Captura de pantalla de ArcGIS Pro con los datos de entrenamiento de [Esri](#)

19

Coordenadas y elevación

Todas las geometrías de CityGML deben utilizar valores de coordenadas 3D. Cada punto 3D está absolutamente georreferenciado. Esto significa que todas las coordenadas pertenecen a un sistema de referencia de coordenadas (CRS) y no se permiten transformaciones locales.

Esto contrasta con BIM, donde el uso de coordenadas locales es común.

CityGML puede incluir la representación de modelos digitales del terreno (MDT), por ejemplo, como nubes de puntos sobre datos ráster o TIN. El concepto LOD permite incluso combinar varias variantes de terreno en diferentes resoluciones. Puede ser, por ejemplo, un TIN de alta resolución incrustado en un DTM cuadrículado de un área grande.

A veces uno se encuentra con el término 2.5D de datos. En ese caso, hay la misma Z para cada posición (X, Y). Estos modelos 2.5D mantienen los costos bajos en comparación con los modelos 3D, siendo aún más útiles para los análisis espaciales en comparación con los datos 2D.

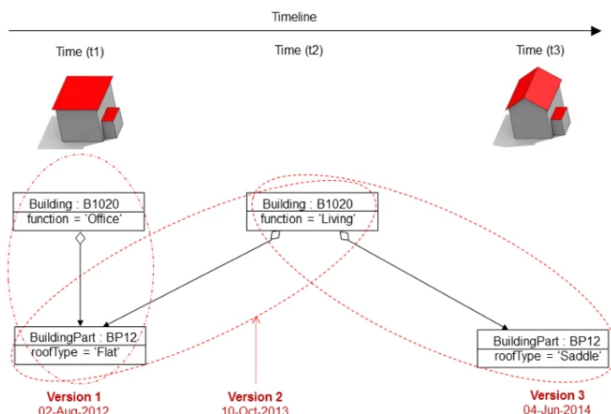


Estándares de datos 3D

4D – Dimensión Temporal

- Importante en las ciudades inteligentes, los gemelos digitales
- Módulo de control de versiones: cambios lentos
- Módulo dinamizador: cambios rápidos, p. ej., datos de sensores

```
<cityObjectMember>
  <Building gml:id="B1020_t1">
    <identifier>B1020</identifier>
    <consistsOfBuildingPart>
      <BuildingPart xlink:href="//identifier(text()='BP12')"/>
    </consistsOfBuildingPart>
    <creationDate>2012-08-02</creationDate>
    <terminationDate>2013-10-10</terminationDate>
    <function>Office</function>
  </Building>
</cityObjectMember>
<cityObjectMember>
  <Building gml:id="B1020_t2">
    <identifier>B1020</identifier>
    <consistsOfBuildingPart>
      <BuildingPart xlink:href="//identifier(text()='BP12')"/>
    </consistsOfBuildingPart>
    <creationDate>2013-10-10</creationDate>
    <function>Living</function>
  </Building>
</cityObjectMember>
```



Ejemplo de versiones que representan modificaciones de un edificio (up)
Representación de diferentes versiones de objetos de ciudad dentro de un
conjunto de datos CityGML codificado en GML (izquierda)
De: Kutzner et al (2020) CityGML 3.0: Las nuevas funciones abren nuevas
aplicaciones

20

4D – dimensión temporal

La dimensión temporal es cada vez más importante en aplicaciones como las ciudades inteligentes y los gemelos digitales.

Las propiedades dependientes del tiempo son administradas por el módulo de control de versiones y el módulo dinamizador, dos de los módulos azules de aspecto específico en el modelo conceptual (Fig. XX).

El módulo Control de versiones gestiona los cambios cualitativos que son de naturaleza más lenta.

Todas las entidades pueden tener los atributos creation-Date y termination-Date, así como validFrom y validTo. Estos representan una versión específica de un objeto en un momento dado. Por lo tanto, el conjunto de datos CityGML se puede consultar para saber cómo se ve el modelo de ciudad en un momento específico.

El módulo Dynamizer gestiona los cambios cuantitativos con variaciones frecuentes o dinámicas de las propiedades de los objetos. Pueden ser observaciones de sensores en tiempo real, datos de dispositivos IoT o simulaciones. Algunos ejemplos son la densidad del tráfico, la contaminación del aire, el consumo de electricidad, la irradiación solar o la posición de objetos en movimiento.

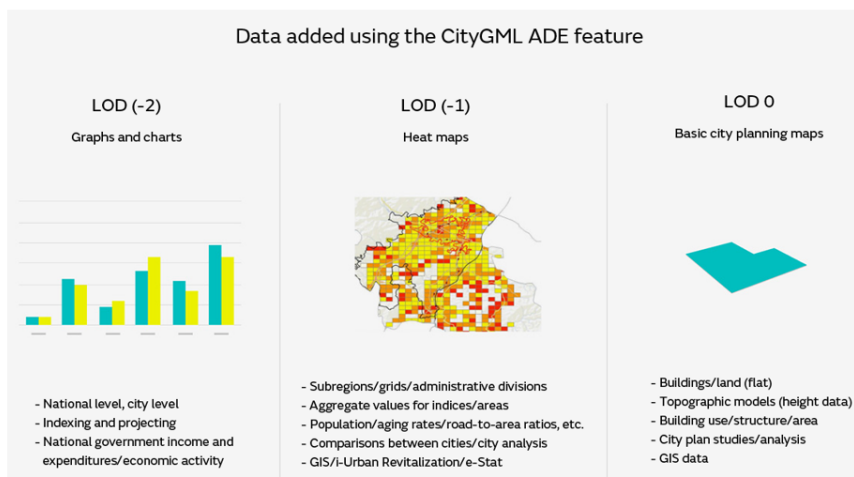
En este caso, solo algunas de las propiedades de los objetos estáticos deben tener valores variables en el tiempo.



Estándares de datos 3D



Extensiones de dominio de aplicación, ADE



- Los ADE facilitan la adición de nuevas clases, atributos o relaciones
- por ejemplo, ADE de energía
- ADE de red de servicios

[3d-urban-models-04en.png \(1200x650\) \(murata.com\)](#)

21

Extensiones de dominio de aplicación, ADE en CityGML

El modelo de datos principales de CityGML 3.0 se puede ampliar mediante ADE, Application Domain Extensions.

Los ADE facilitan la extensión sistemática del modelo conceptual CityGML mediante nuevas clases, atributos y relaciones para dominios de aplicación específicos.

Es decir, los ADE se usan cuando los profesionales desean modelar características adicionales y cuando GenericCityObjects y los atributos genéricos no son lo suficientemente sistemáticos para el propósito.

Se han desarrollado muchos ADE para diversas aplicaciones; por ejemplo, el ADE de Energía para apoyar los análisis energéticos de edificios o el ADE de Red de Servicios Públicos para la representación y análisis de múltiples redes.

Con CityGML 3.0, los ADE se convierten en modelos independientes de la plataforma a nivel conceptual. Como tales, se pueden combinar libremente con las clases originales del modelo conceptual.

Biljecki et al. (2018) CityGML Application Domain Extension (ADE): overview of developments, Open Geospatial Data, Software and Standards, DOI: [10.1186/s40965-018-0055-6](https://doi.org/10.1186/s40965-018-0055-6) proporcionan un análisis exhaustivo de los ADE CityGML existentes.

S

Estándares de datos 3D



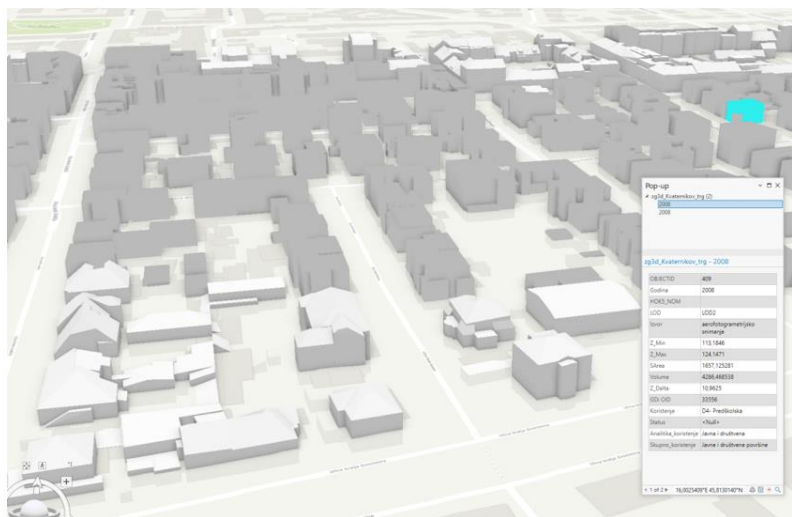
Co-funded by
the European Union

Calidad de los datos

Los modelos de ciudad son grandes conjuntos de datos complejos

La calidad de los datos es un tema crucial

- Exactitud
- Integridad
- Usabilidad
- Consistencia
- Unicidad



Modelo semántico 3D de Zagreb, Croacia. Captura de pantalla de ArcGIS Pro.

22

S

Calidad de los datos

Los modelos de ciudades son muy complejos, con enormes conjuntos de datos y mucha información procedente de múltiples fuentes. Por lo tanto, la calidad de los datos es una cuestión crucial a tener en cuenta. De los malos datos surgen malas decisiones.

Hay varias dimensiones de la calidad de los datos. Estas dimensiones pueden tener pesos iguales o variables, pero todas ellas deben estar cubiertas.

En el caso de los datos espaciales, las principales dimensiones de calidad son la precisión y la integridad.

Precisión posicional: ¿Las coordenadas geográficas de una entidad se corresponden con las coordenadas del objeto en el mundo real? Dentro del mismo sistema de referencia, por supuesto.

Precisión temática: ¿Los datos representan correctamente las clasificaciones asociadas con ubicaciones u objetos? Por ejemplo, ¿es un río en el modelo realmente un río en la realidad?

Precisión temporal: ¿Están los datos actualizados y representan la realidad durante un período de tiempo requerido?

Precisión topológica: ¿La relación espacial entre las entidades refleja correctamente su posición en el mundo real?

Integridad: Son todos los puntos de datos disponibles en la base de datos; ¿Faltan muchos valores para los datos necesarios? Podemos añadir incluso otras dimensiones como:

Usabilidad: ¿Los datos se alinean con las necesidades del usuario final? ¿Es el nivel de detalle apropiado para el propósito?

Coherencia: ¿Están sincronizados los datos en todas las fuentes de datos? ¿Hay valores no posibles?
Unicidad: ¿hay duplicados en los datos? ¿Todo el mundo utiliza el mismo conjunto de datos?

Estándares de datos 3D

S

Codificaciones de CityGML

GML – Lenguaje de Marcado Geográfico

CityGML es el nombre de:

- Codificación GML basada en XML modelo de datos conceptual
- Emitido por Open Geospatial Consortium (OGC)

CityGML 3 permite codificar los datos en XML, en JSON o en esquemas de base de datos

Three encodings:

- XML-based →



- JSON-based →



- SQL-based →



23

S

Codificaciones de CityGML

En primer lugar, debemos tener en cuenta que CityGML es el nombre tanto para la codificación GML como para el modelo de datos.

GML se queda para la [versión 3.1.1 \(GML3\) del lenguaje de marcado geográfico](#). Es el estándar internacional extensible para el intercambio de datos espaciales emitido por el [Open Geospatial Consortium \(OGC\)](#) y la [ISO TC211](#), que permite un acceso fácil y gratuito a toda la comunidad internacional.

CityGML 3 permite codificar los datos en GML/XML, pero también en JSON o esquemas de base de datos.



Codificación XML

Codificación original de
CityGLM

- verboso
- jerárquico
- complejo
- No adaptado para la web

Ya no se usa mucho

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <CityModel xmlns:xlink="http://www.w3.org/1999/xlink"
3   xmlns:gml="http://www.opengis.net/gml"
4   xmlns="http://www.opengis.net/citygml/2.0"
5   xmlns:bldg="http://www.opengis.net/citygml/building/2.0"
6   xsi:schemaLocation="http://www.opengis.net/citygml/2.0">
7   <cityObjectMember>
8     <bldg:Building gml:id="9a06451677c7">
9       <bldg:function>1070</bldg:function>
10      <bldg:lodSolid>
11        <gml:Solid>
12          <gml:exterior>
13            <gml:CompositeSurface>
14              <gml:surfaceMember>
15                <gml:Polygon>
16                  <gml:exterior>
17                    <gml:LinearRing>
18                      <gml:pos>0.0 0.0 0.0</gml:pos>
19                      <gml:pos>0.0 1.0 0.0</gml:pos>
20                      <gml:pos>1.0 1.0 0.0</gml:pos>
21                      <gml:pos>1.0 0.0 0.0</gml:pos>
22                      <gml:pos>0.0 0.0 0.0</gml:pos>
23                    </gml:LinearRing>
24                  </gml:exterior>
25                </gml:Polygon>
26              </gml:surfaceMember>
27            ...
28          </bldg:Building>
29          <bldg:Building gml:id="jdhd76sa">
30            ...
31          </bldg:Building>
32        </cityObjectMember>
33      </CityModel>
```

24

Codificación XML

Originalmente, CityGML eran archivos XML. En la figura se da un ejemplo. Como podemos ver, la codificación XML es detallada, jerárquica, compleja y no está adaptada para la web. Por lo tanto, ya no se usa mucho (en CityGML 3).



Codificación CityJSON II

- la alternativa más utilizada a la codificación XML JSON - JavaScript Object Notation
- incluso CityJSON es un estándar OGC
- Las coordenadas se almacenan en un solo lugar, en una matriz separada, es decir, los "vértices"

CityJSON permite la compresión completa y simplifica la estructura de archivos, en comparación con la codificación XML (figura derecha)
Ejemplo de coordenadas en CityJSON (figura izquierda)

```
1  "vertices": [  
2    [23234, 111009, 1392],  
3    [29456, 115134, 1007],  
4    [54508, 229995, 1961],  
5    ...  
6    [23134, 625134, 203]  
7  ]
```

```
1  "CityObjects": {  
2    "id-1": {  
3      "type": "Building",  
4      "attributes": {...},  
5      "children": ["id-2", "id-3"],  
6      "geometry": [{...}]  
7    },  
8    "id-2": {  
9      "type": "BuildingPart",  
10     "parents": ["id-1"],  
11     "geometry": [{...}]  
12     ...  
13   },  
14   "id-3": {  
15     "type": "BuildingPart",  
16     "parents": ["id-1"],  
17     "geometry": [{...}]  
18     ...  
19   }  
20 }
```

25

Codificación CityJSON II

Actualmente, la alternativa más utilizada a la codificación XML de CityGML es CityJSON (JavaScript Object Notation).

Incluso CityJSON es un estándar del Open Geospatial Consortium (OGC). La versión actual de CityJSON es la 1.1.3 y proporciona una serie de ventajas sobre CityGML-XML.

[Especificaciones de CityJSON 1.1.3](#)

En primer lugar, las coordenadas se almacenan en un solo lugar, en una matriz separada, es decir, la propiedad "vértices" del objeto CityJSON. Las primitivas geométricas se refieren a la posición de un vértice en la matriz.



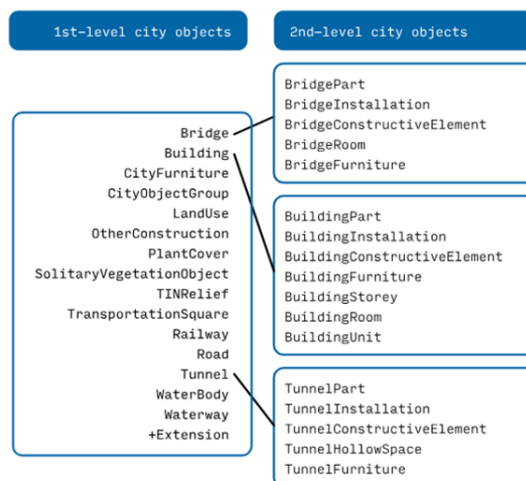
Estándares de datos 3D

Codificación CityJSON II

CityJSON:

- Es muy adecuado para aplicaciones web
- Reduce el tamaño de los datos; El archivo JSON ocupa aproximadamente 6 veces menos espacio que XML
- Se puede almacenar tanto en bases de datos relacionales como en bases de datos no-SQL

Dos tipos de objetos de la ciudad: 1º y 2º nivel (padres e hijos)



[CityJSON Specifications 1.1.3](#)

26

Codificación CityJSON II

Luego, JSON domina la web, lo que hace que el intercambio de datos entre aplicaciones sea una tarea fácil.

En tercer lugar, JSON reduce el tamaño de los datos; El archivo JSON ocupa aproximadamente 6 veces menos espacio que XML.

Por último, los archivos JSON se pueden almacenar tanto en bases de datos relacionales como en bases de datos no SQL.

En JSON, el esquema de CityGML se ha aplanado en comparación con la codificación XLM, como se muestra en la figura.

Hay dos tipos de objetos de ciudad en CityJSON: de 1er nivel, o padres, que pueden existir por sí mismos, como un edificio, un puente o un cuerpo de agua.

Y 2º nivel, o niños, que necesitan tener un "padre" para existir. Algunos ejemplos pueden ser la parte del edificio, los muebles o la habitación. En la figura se muestran más ejemplos. Los objetos de ciudad de 1º y 2º nivel se almacenan en el diccionario "CityObjects".

Un objeto de ciudad debe tener, como mínimo, una propiedad de "geometría". Si se van a almacenar atributos, tienen que estar en la propiedad "atributos". Esto simplifica el trabajo del desarrollador porque hay un único punto de entrada para todas las geometrías y atributos (a diferencia de CityGML codificado en XML).

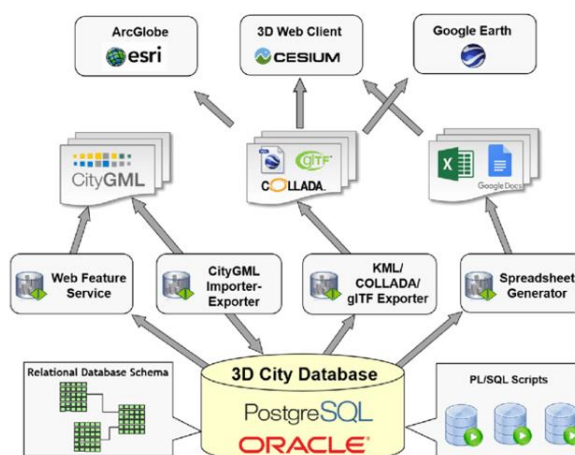
Para explorar un modelo CityJSON simple, vaya a: <https://ninja.cityjson.org/> que es un visor JSON en línea gratuito y cargue el LoD3_Railway.city.json, disponible con la clase. Alternativamente, puede descargarlo en: [Conjuntos de datos | CityJSON](#), junto con otros archivos .json.

Estándares de datos 3D

Codificación de bases de datos de ciu

- Esquema de base de datos llamado 3DCityDB
- No es estándar oficial
- Incluso software de código abierto

[3DCityDB Database – Homepage](#)



3DCityDB: una solución de geodatabase 3D para la gestión, el análisis y la visualización de modelos semánticos de ciudades en 3D basados en CityGML, Yao et al. (2018)

27

Codificación de bases de datos de ciudades 3D

La tercera codificación CityGML es un esquema de base de datos llamado 3DCityDB. Se puede implementar tanto para PostgreSQL como para Oracle Spatial. Es un software de código abierto y aunque no es un estándar oficial, es utilizado por varios municipios de todo el mundo.

Un paquete de software ampliamente utilizado para tratar con modelos 3D se llama 3D City Database, abreviado como 3DCityDB. Es un esquema de base de datos gratuito y de código abierto para sistemas de gestión de bases de datos relacionales espacialmente mejorados, como ORACLE Spatial o PostgreSQL/PostGIS.

Además del sistema de base de datos, 3DCityDB proporciona todas las herramientas necesarias, incluido el mantenimiento y la visualización web de los modelos de ciudades en 3D.

Para tutoriales / curso ver

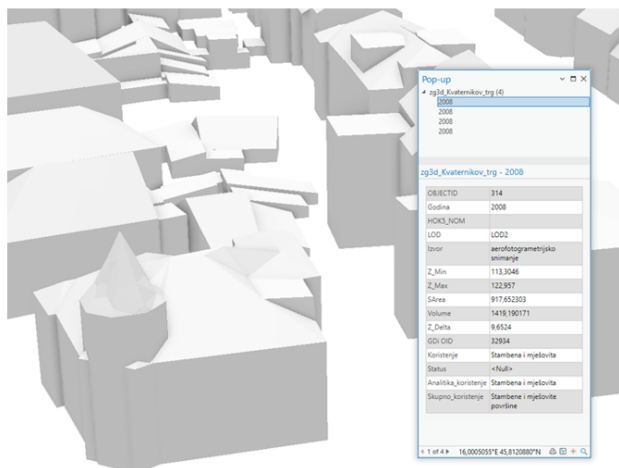
http://www.3dcitydb.net/3dcitydb/fileadmin/TUM_Workshop/Documents/Tutorial.pdf

Estándares de datos 3D

Otros formatos 3D

GML se puede combinar con muchos otros formatos

- Web Feature Service (WFS)
- Web Processing Service (WPS)
- KML/COLLADA or X3D files
- Web 3D Service (W3DS)
- Web Terrain Service (WTS)
- Indoor GLM



Ejemplo de un modelo 3D

28

Otros formatos 3D

Como estándar OGC, GML3 se puede combinar con la gama completa de otros estándares OGC. El servicio de entidades web (WFS), el servicio de catálogo (CS-W), el servicio de transformación de coordenadas web (WCTS) y el servicio de procesamiento web (WPS) son especialmente relevantes para acceder, procesar e identificar los recursos de CityGML.

Para la visualización en 3D, CityGML es un formato base del que se pueden derivar fácilmente los formatos gráficos en 3D. La rica información semántica de los objetos CityGML también ayuda en la generalización y simbolización cartográfica automática. También es adecuado para generar gráficos por ordenador representados, por ejemplo, como archivos KML/COLLADA o X3D. Los servicios de representación de OGC correspondientes son el Servicio Web 3D (W3DS) y el Servicio Web Terrain (WTS).

También hay Indoor GML para el modelado de datos y la navegación en espacios interiores. Tiene una superposición mínima con CityGML, pero es posible combinar estos dos (CityGML e Indoor GLM) en un solo modelo, si es necesario.



Estándares de datos 3D

LandInfra

- Otro estándar de datos 3D
- Ingeniería terrestre y civil
- Algunos se solapan con CityGML
- Incluye funciones no disponibles en CityGML



[Comparison between IFC, CityGML and LandInfra. Source: LandInfra BIM_GIS.pdf](#)

29

LandInfra

Además de CityGML, existe otro estándar de datos 3D llamado LandInfra.

Es un modelo conceptual para la infraestructura terrestre y de ingeniería civil. Como tal, se publica para casos de uso predeterminados, como instalaciones, proyectos, carreteras, ferrocarriles, topografía, división de tierras, aguas residuales o sistemas de distribución de agua.

LandInfra tiene cierta superposición potencial con CityGML, pero no es compatible con ADE y LOD.

Por otro lado, incluye algunas características que no están disponibles en CityGML. Estos incluyen: modelado mejorado de datos del subsuelo, marco para modelar información legal y almacenamiento de información relacionada con la encuesta.

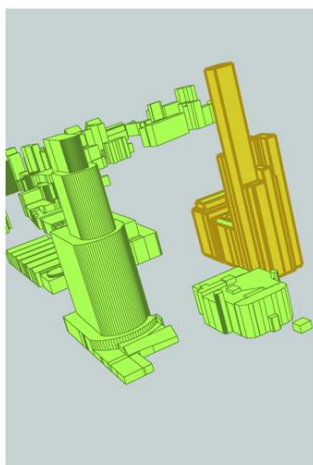
Incluso se superpone parcialmente con .ifc, pero aún no se utiliza para cerrar la brecha entre BIM y GIS.

Para obtener más información, puede leer el artículo [LandInfra BIM_GIS.pdf](#)

Influenciado por CityGLM,
pero simplificado

Objetivo de garantizar la interoperabilidad de los datos y servicios espaciales de los diferentes países de la UE

Requisitos de las directivas de la UE (por ejemplo, ruido, rendimiento energético)



Property	Value
Type	BuildingPart
Feature System	EPIC20092
Dimension	3D
Number of Vertices	8200
Min Edents	93529.243000000000, 435451.2372, 0
Max Edents	93529.243000000000, 435451.237200000000, 135.500084000000
Attributes (12)	
beginGeometry (encoded utf-8)	2013-01-15T00:00:00
conditionOfConstructionArea (encoded utf-8)	True
conditionOfConstructionArea_nil (encoded utf-8)	True
fine.geometry (string)	fine_geometry
fine.geometry (string)	fine_area
geometryIdOfBuildingGeometryOfCulc2 (horizontalGeometry_nul (encoded utf-8))	1.0
geometryIdOfBuildingGeometryOfCulc2 (horizontalGeometry_nul (encoded utf-8))	fine-geom-778111-1978-4cub-a524-c52524252
geom_id (encoded utf-8)	fine-geom-778111-1978-4cub-a524-c52524252
geom_idOfBuildingArea (encoded utf-8)	fine-geom-778111-1978-4cub-a524-c52524252
encoded_IsolateNameSpace (encoded utf-8)	EPIC20092
unit.type (string)	unit_area
IFME:Macros (122 Parts)	
Name (encoded utf-8)	geometryIdOfC2BuildingGeometryOfCulc2.geometryMacros
Geometry Trads (1)	
Part 0: IFMPPolygon	
Name (encoded utf-8)	surfaceMember
Geometry Trads (1)	
gm_id (encoded utf-8)	fine-geom-778111-1978-4cub-a524-c52524252-1
Linear Boundary	
Convex	True
Orientation	Right Hand Rule
Boundary IFMLine (4 Coordinates)	93529.243000000000, 435451.237200000000, 135.500084000000, 0
Closed	Yes
Coordinates (4)	93529.243000000000, 435451.237200000000, 135.500084000000, 0
C1	93529.243000000000, 435451.237200000000, 135.500084000000
C2	93529.243000000000, 435451.237200000000, 135.500084000000
C3	93529.243000000000, 435451.237200000000, 135.500084000000
C4	93529.243000000000, 435451.237200000000, 135.500084000000
Part 1: IFMPoint	
Name (encoded utf-8)	True
Geometry Trads (1)	fine-geom-778111-1978-4cub-a524-c52524252-2
Linear Boundary	True
Orientation	Right Hand Rule
Boundary IFMLine (5 Coordinates)	93529.243000000000, 435451.237200000000, 135.500084000000, 0

INSPIRE Buildings GML visto con el Inspector de datos. De: [Converting CityGML to INSPIRE 3D Buildings \(Annex III\) \(safe.com\)](#)

30

Inspira edificios en 3D

CityGML ha influido fuertemente en el modelo de INSPIRE 3D Buildings.

INSPIRE GML incluye atributos para respaldar el mandato de INSPIRE, muchos de los cuales son comunes a otros temas de INSPIRE.

INSPIRE Buildings está algo simplificado, en comparación con CityGML. Por ejemplo, no se requieren ventanas ni puertas, las piezas no pueden tener subpartes y las apariencias se simplifican.

La especificación de datos de INSPIRE no requiere la recopilación de nuevos datos. Su objetivo es garantizar la interoperabilidad de los datos espaciales y los servicios de diferentes países de la UE. Esto aborda los requisitos relacionados con la presentación de informes europeos, como la Directiva sobre el ruido, la Directiva sobre la calidad del aire o la Directiva sobre la eficiencia energética de los edificios.



Referencias

Malhotra et al. (2022) Modelado de información para la simulación energética de edificios urbanos: una revisión taxonómica. Edificación y Medio Ambiente 208,
<https://doi.org/10.1016/j.buildenv.2021.108552>

Ledoux (2018) CityGML y sus dos codificaciones CityGML y CityJSON, presentación Universidad de Delft

Ken Arroyo Otori, Hugo Ledoux y Ravi Peters (2020-2022): Modelado 3D del entorno construido, disponible en: [Lanzamientos · tudelft3d/3dbook \(github.com\)](#)

Sahleb (2019) Conversión automática de CityGML a IFC, TU Delft, tesis de maestría