10th Int. CityGML Energy ADE workshop

CityGML & Energy ADE crash course

Giorgio Agugiaro & Kavisha Kumar

TU Delft, 5 December 2018

Greetings from your trainers!

Dr. Giorgio Agugiaro

g.agugiaro@tudelft.nl

https://3d.bk.tudelft.nl/gagugiaro

Kavisha Kumar

k.kavisha@tudelft.nl

https://3d.bk.tudelft.nl/kavisha

Outline

This morning

- Part 1A (Giorgio)
  - A gentle introduction to CityGML
  - UML in a nutshell: reading and understanding class diagrams
  - CityGML modules: Core, Building, CityObjectGroup, Generics
- Part 1B (Kavisha)
  - An overview of tools to work with CityGML (with examples)

This afternoon

- Part 2A (Giorgio)
  - Introduction to the Energy ADE
  - Energy ADE structure and modules
  - A quick mention to the Utility Network ADE
- Part 2B (Kavisha)
  - An overview of tools to work with Energy ADE (with examples)

10th Int. CityGML Energy ADE workshop

A gentle introduction to CityGML as open standard for semantic 3D city modelling

Giorgio Agugiaro

TU Delft, 5 December 2019

License

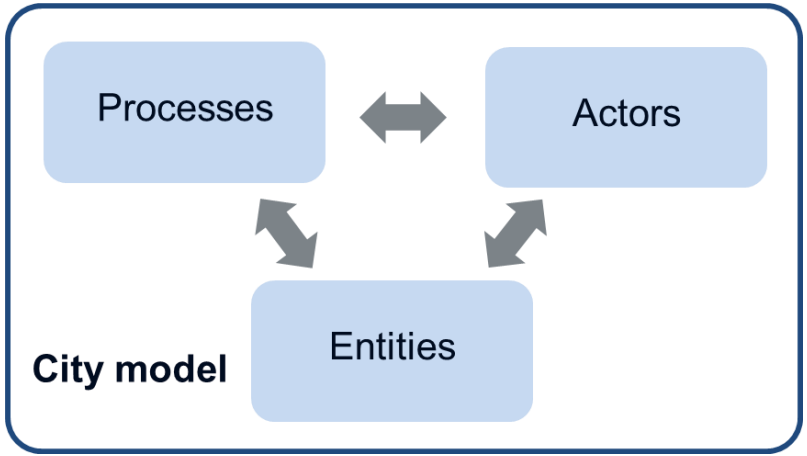
City modelling

Real city

http://media.gettyimages.com/vectors/city-drawing-vector-id523441181?s=170667a

represented by

Digital twin



City modelling: today

- Separate modelling, generally by specific sectors, e.g.
  - Energy { Community, Models, Indicators, Planning }
  - Mobility { Community, Models, Indicators, Planning }
  - Ecology { Community, Models, Indicators, Planning }
  - Economy { Community, Models, Indicators, Planning }
  - Surveying { Community, Models, Indicators, Planning }

Dealing with urban data...

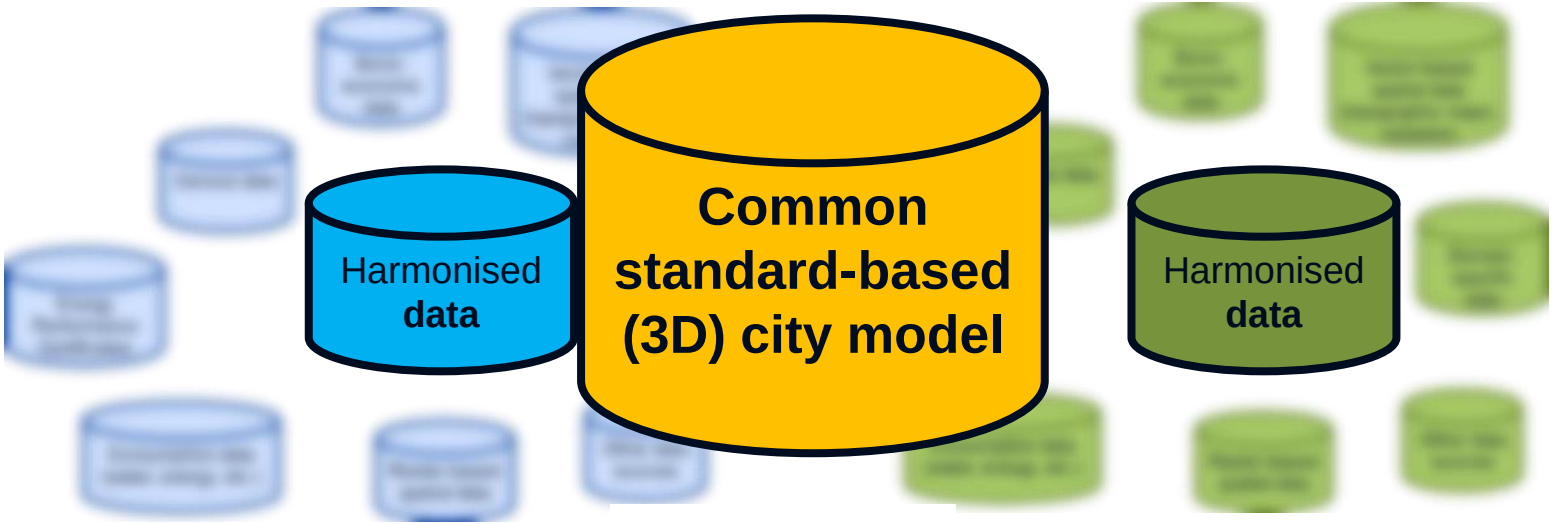
Semantic 3D city modelling

CityGML intro

UML in a nutshell

CityGML reprise

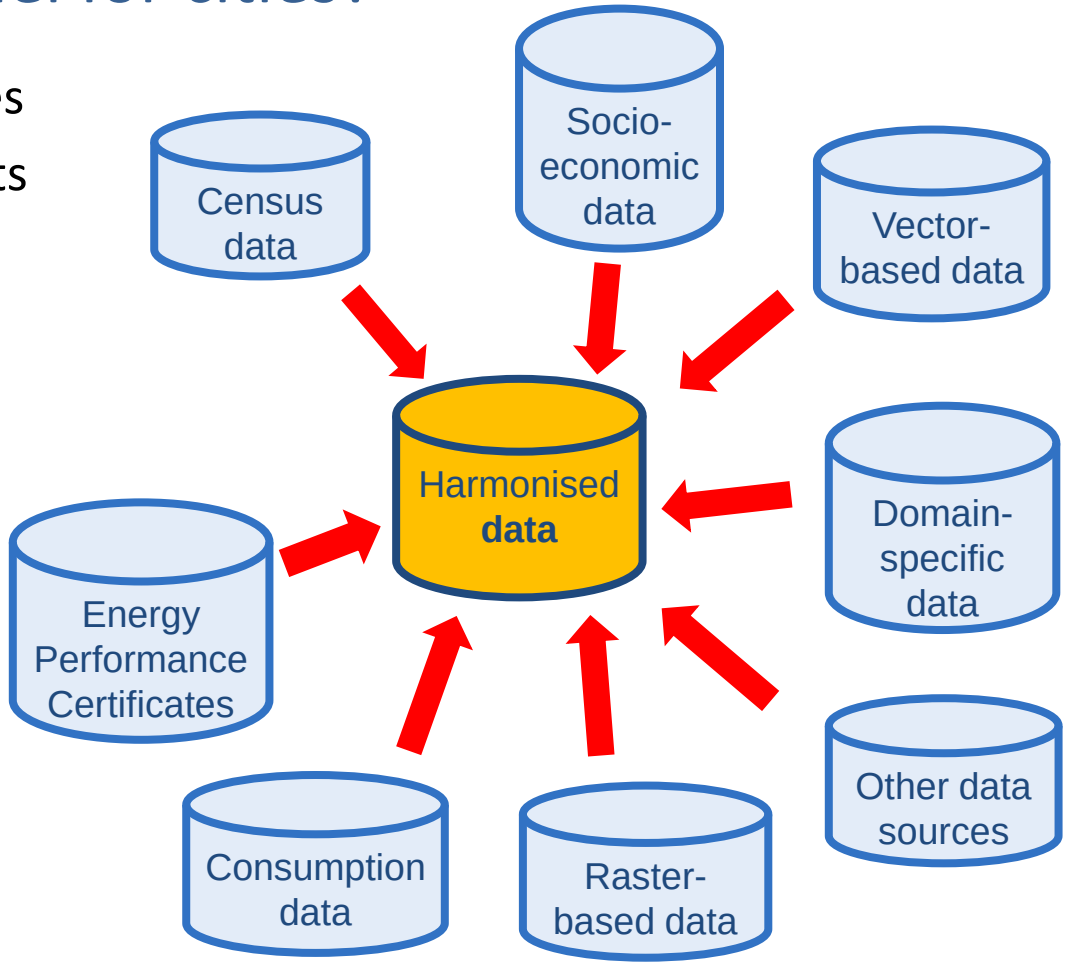
Conclusions



CITY A **CITY A & CITY B** **CITY B**

Which data model for cities?

- Different data sources
- Different data formats
- Different semantics
- Different scales
- Different accuracies
- …



What about standards?

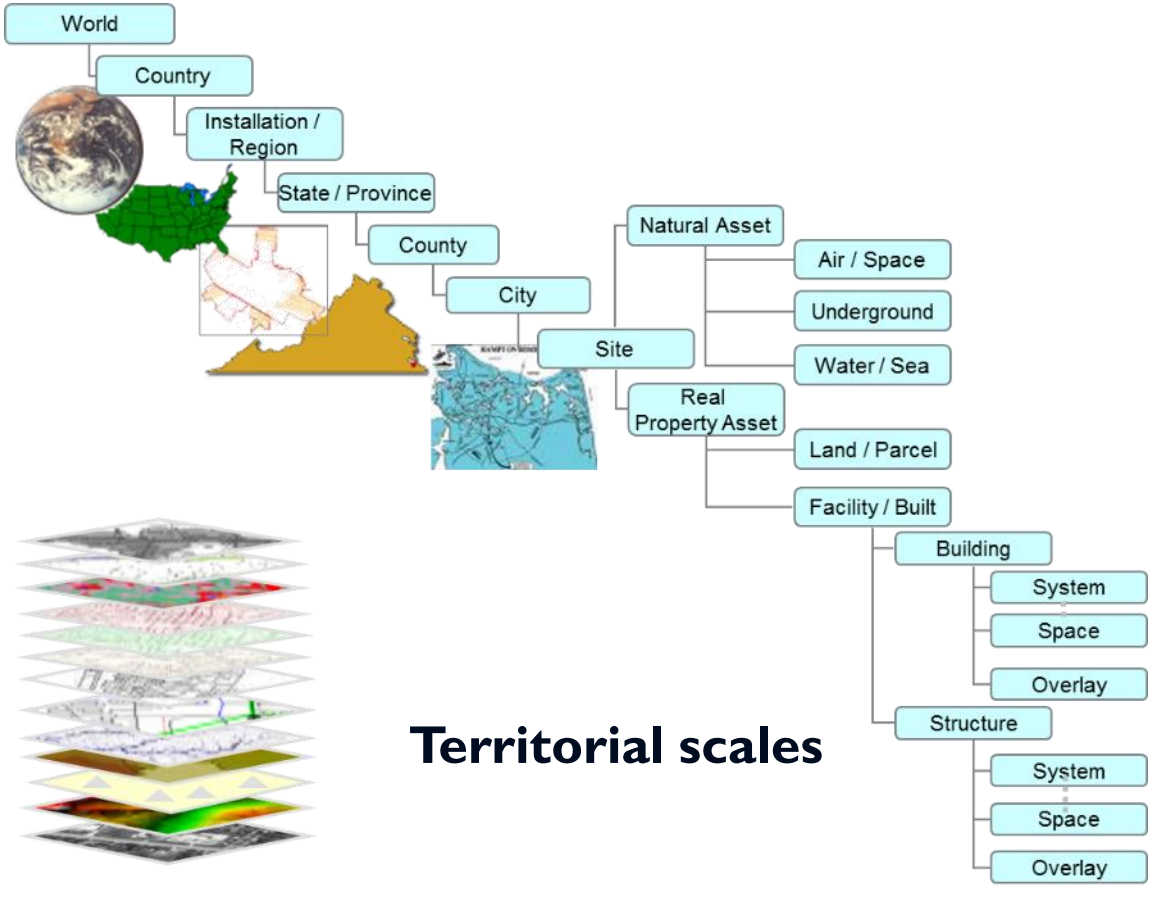
Semantic 3D city modelling

CityGML intro

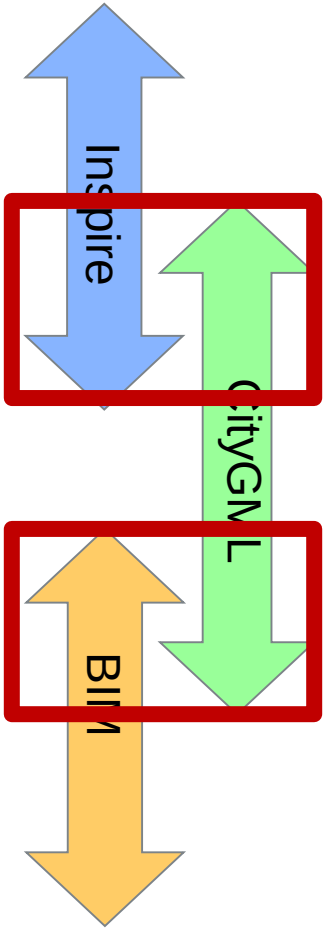
UML in a nutshell

CityGML reprise

Conclusions



Territorial scales



CityGML: City Geography Markup Language

- **Information model** for 3D city models at urban and regional scale (**OGC standard**)
- Comprises **thematic areas** for buildings, terrain, traffic, tunnel, bridges, vegetation, etc.
- Includes multi **level-of-detail 3D geometry**, topology, semantics and appearance
- **Extendible** to other application domains

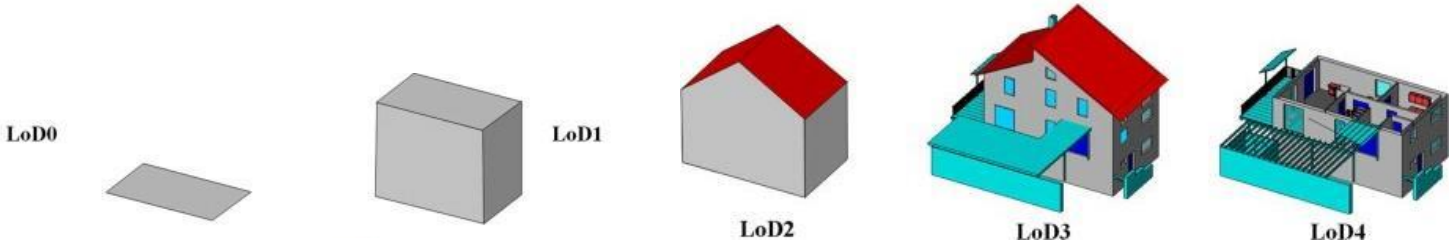


Image source: CityGML 2.0 encoding standard specifications

Semantic 3D city modelling

CityGML intro

UML in a nutshell

CityGML reprise

Conclusions

Image source: Kolbe et al., 2009

Semantic 3D city modelling

CityGML intro

UML in a nutshell

CityGML reprise

Conclusions

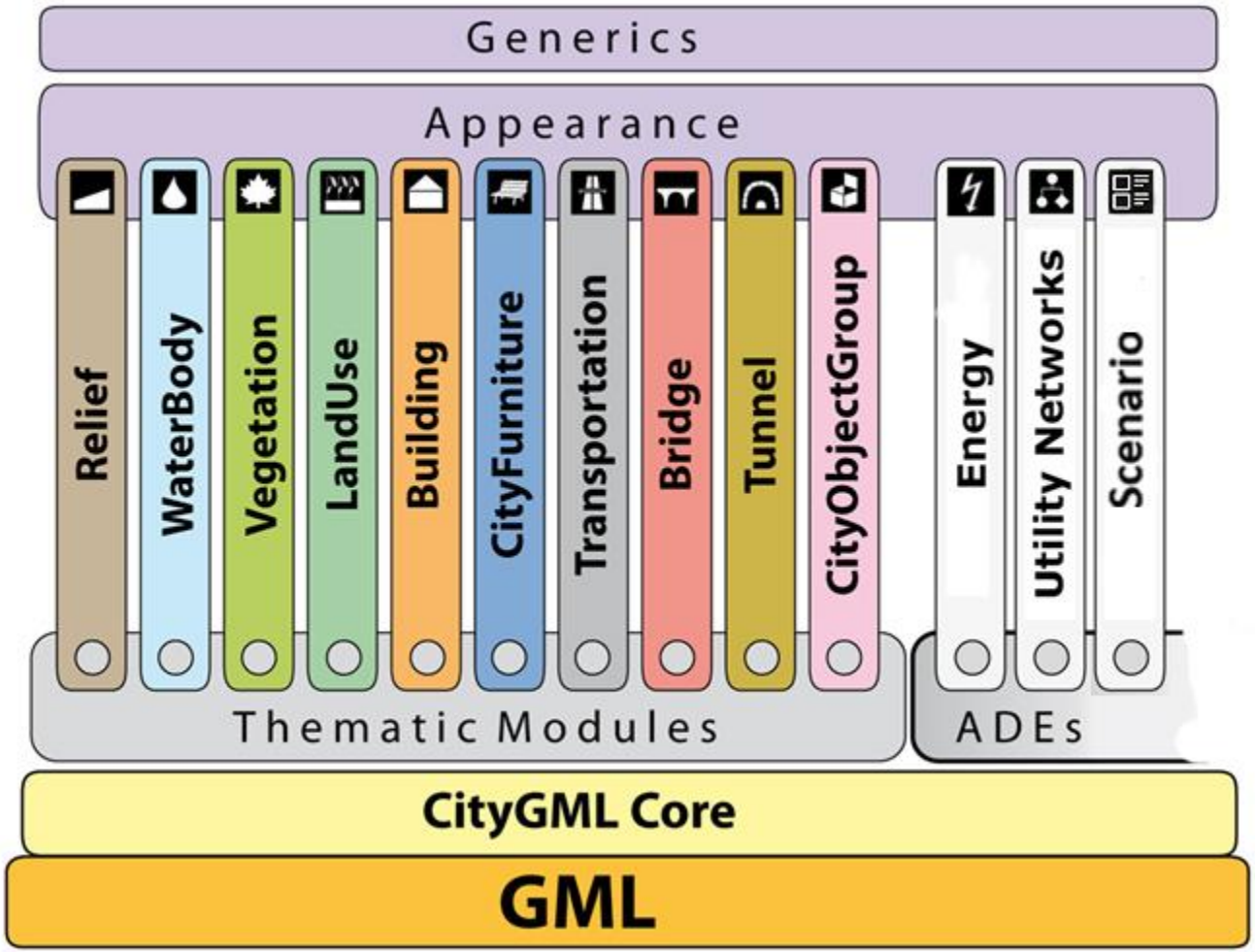


Image source: virtualcitySYSTEMS

CityGML: building model

- Nowadays: creation of 3D city models (up to LoD2) is nearly completely automatic
- Geometric modelling as solids, multi-surfaces, or (from LoD2) thematic surfaces
- Possibility to partition buildings in building parts

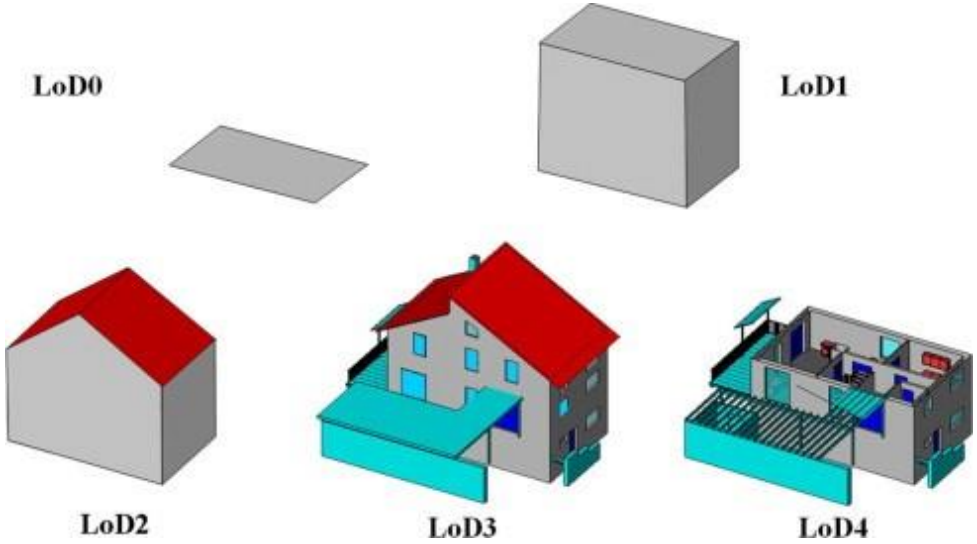


Image source: CityGML 2.0 encoding standard specifications

CityGML: building model

Image source: CityGML 2.0 encoding standard specifications

CityGML: tunnel model

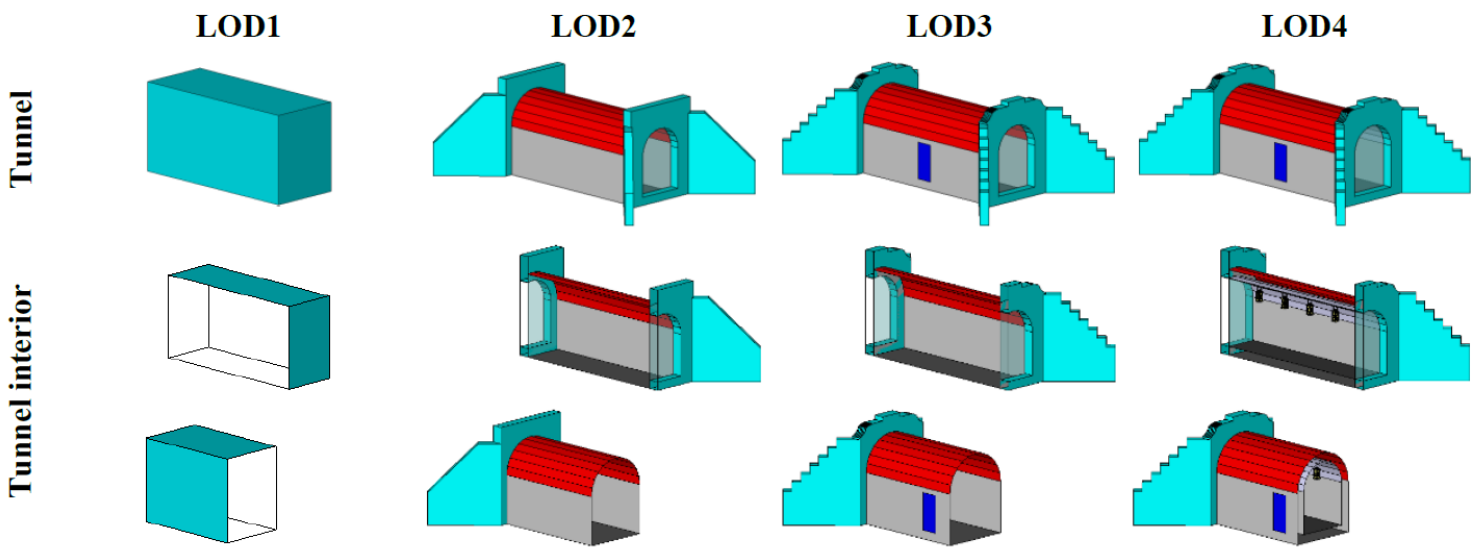


Fig. 40: Tunnel model in LOD1 – LOD4 (source: Karlsruhe Institute of Technology (KIT)).

CityGML: bridge model

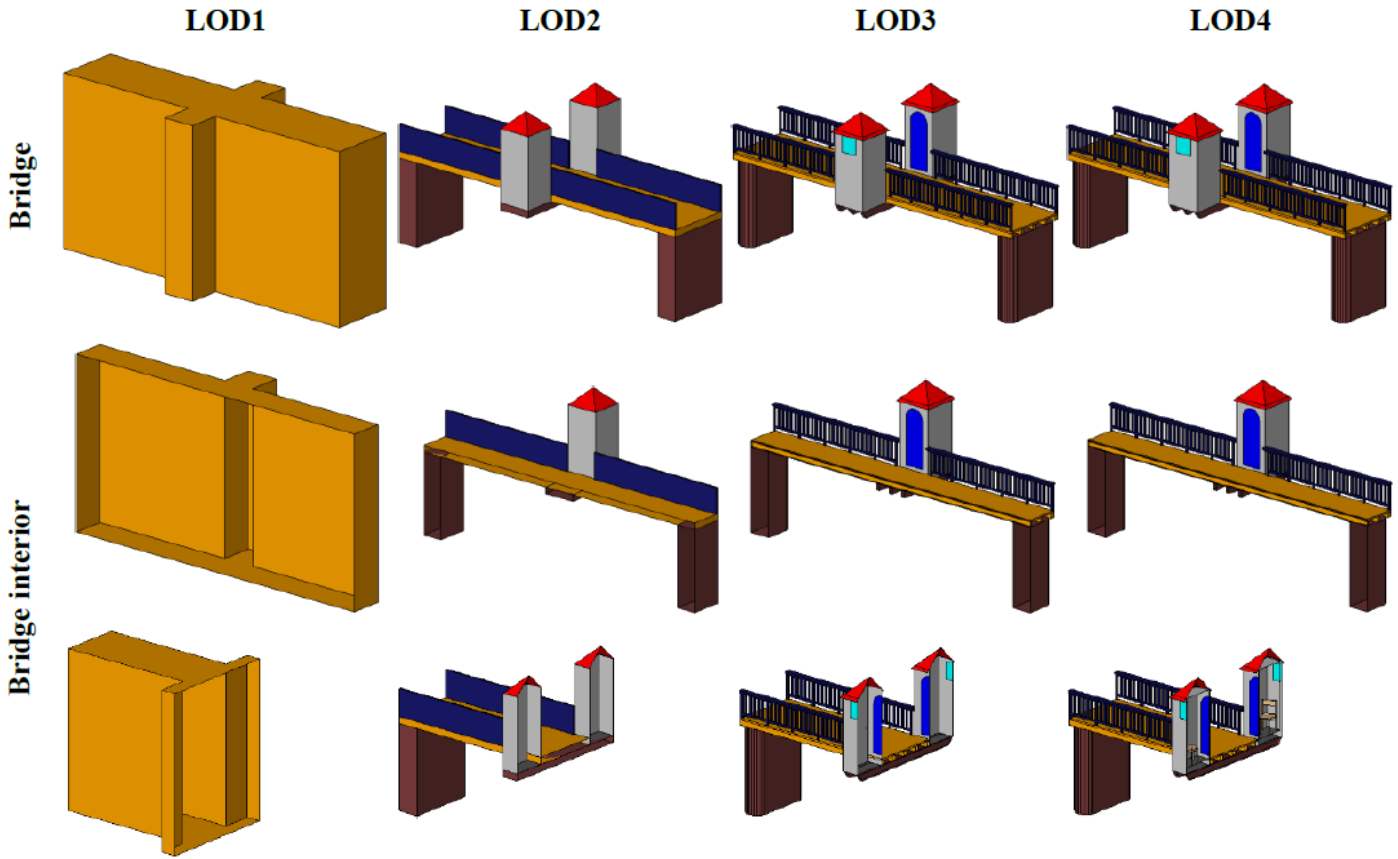


Fig. 46: Bridge model in LOD1 – LOD4. (source: Karlsruhe Institute of Technology (KIT))

CityGML: city furniture model

- Conceived mainly for immovable objects like street lanterns, bus stops, street signs, etc.

- Can be represented also as <u>implicit geometries</u>
  - You use one geometric prototype that you "clone" several times providing each time the specific position, orientation and scaling

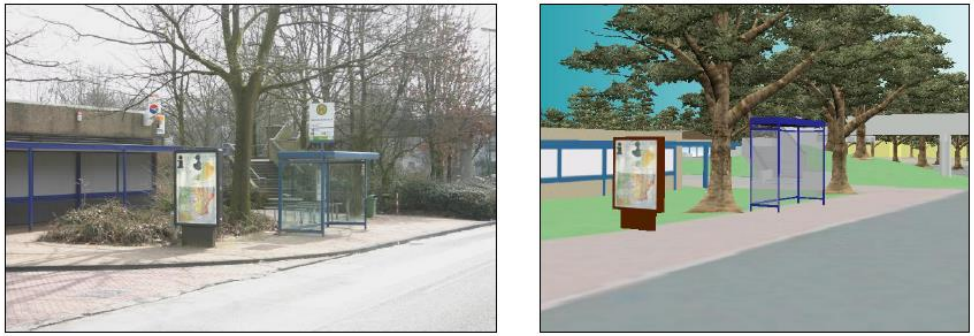


Fig. 67: Real situation showing a bus stop (left). The advertising billboard and the refuge are modelled as *CityFurniture* objects in the right image (source: 3D city model of Barkenberg).

Fig. 68: Real situation showing lanterns and delimitation stakes (left). In the right image they are modelled as *CityFurniture* objects with *ImplicitGeometry* representations (source: 3D city model of Barkenberg).

Image source: CityGML 2.0 encoding standard specifications

CityGML: vegetation model

- Solitary vegetation object can be represented in multiple LoDs with any geometry
- Plant cover can be represented only as MultiSurface or MultiSolid

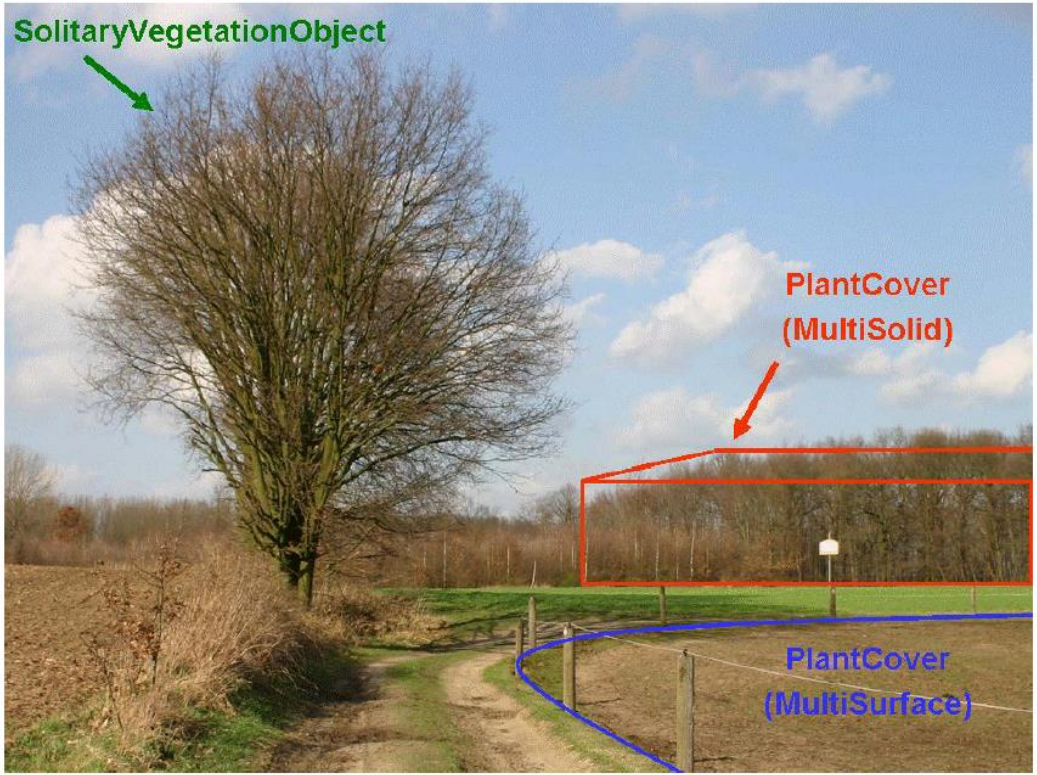


Fig. 63: Example for vegetation objects of the classes *SolitaryVegetationObject* and *PlantCover* (graphic: District of Recklinghausen).

Image source: CityGML 2.0 encoding standard specifications

CityGML: transportation model

Semantic 3D city modelling

CityGML intro

UML in a nutshell

CityGML reprise

Conclusions

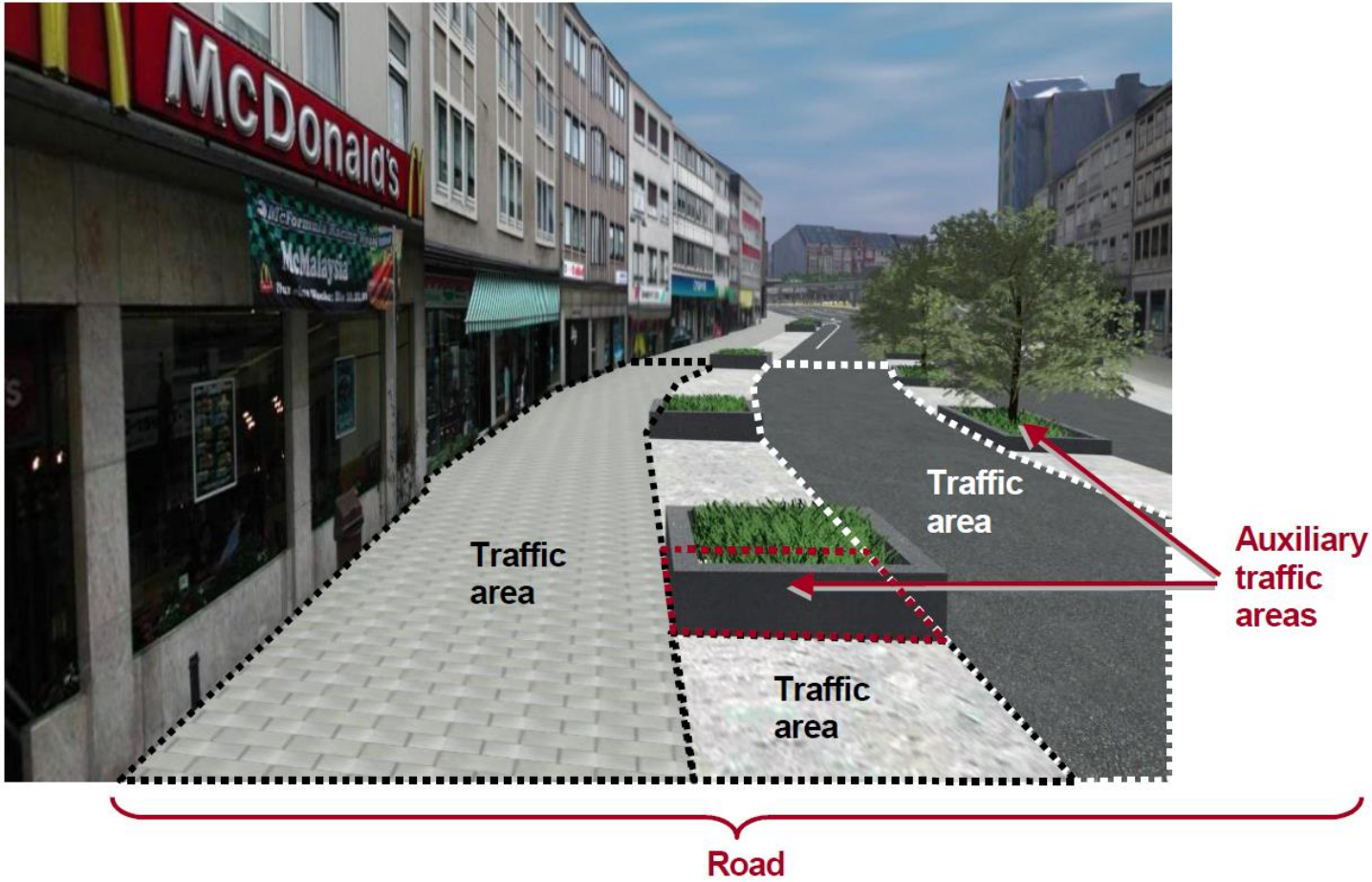


Image source: CityGML 2.0 encoding standard specifications

CityGML: transportation model

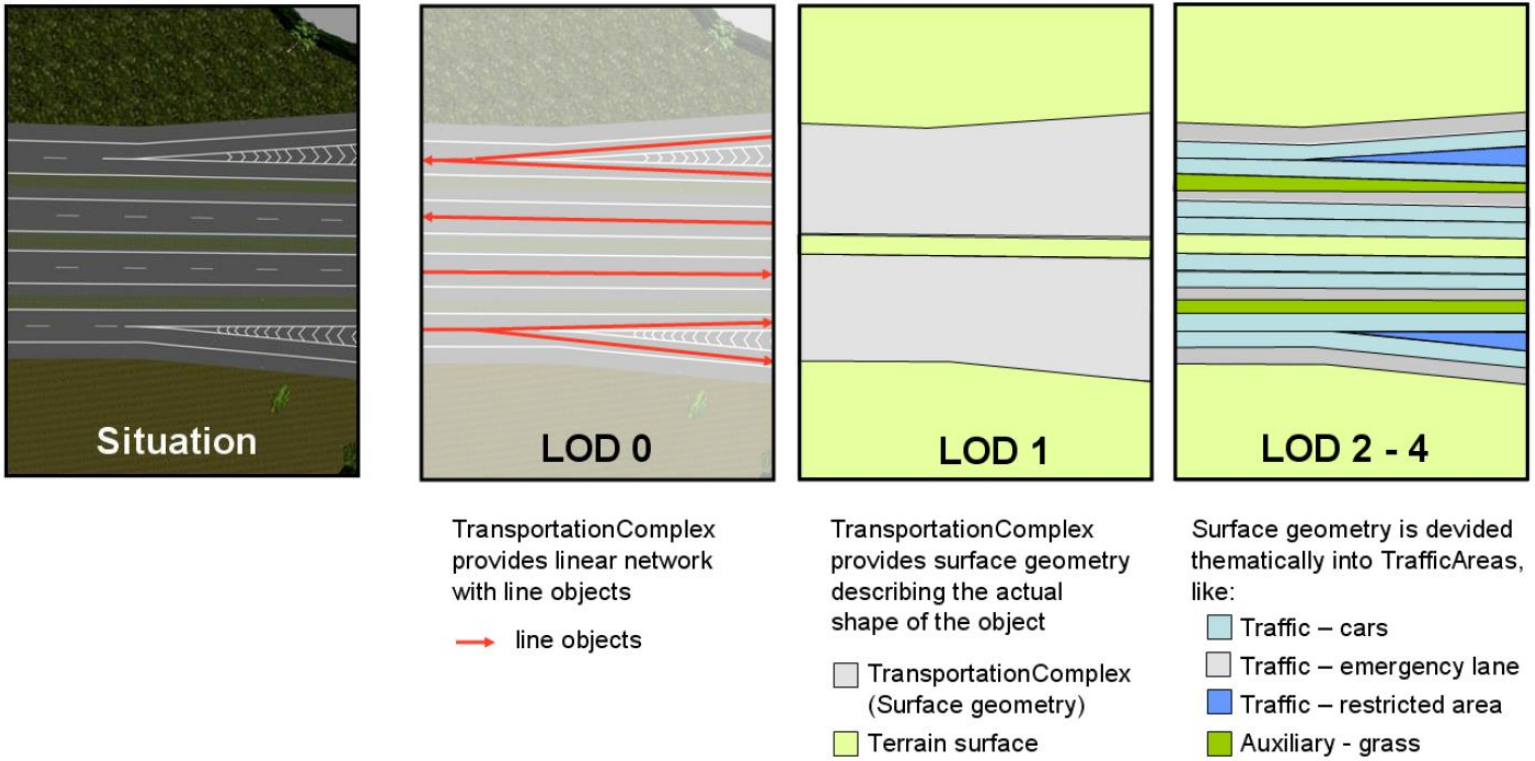


Image source: CityGML 2.0 encoding standard specifications

CityGML: land use model

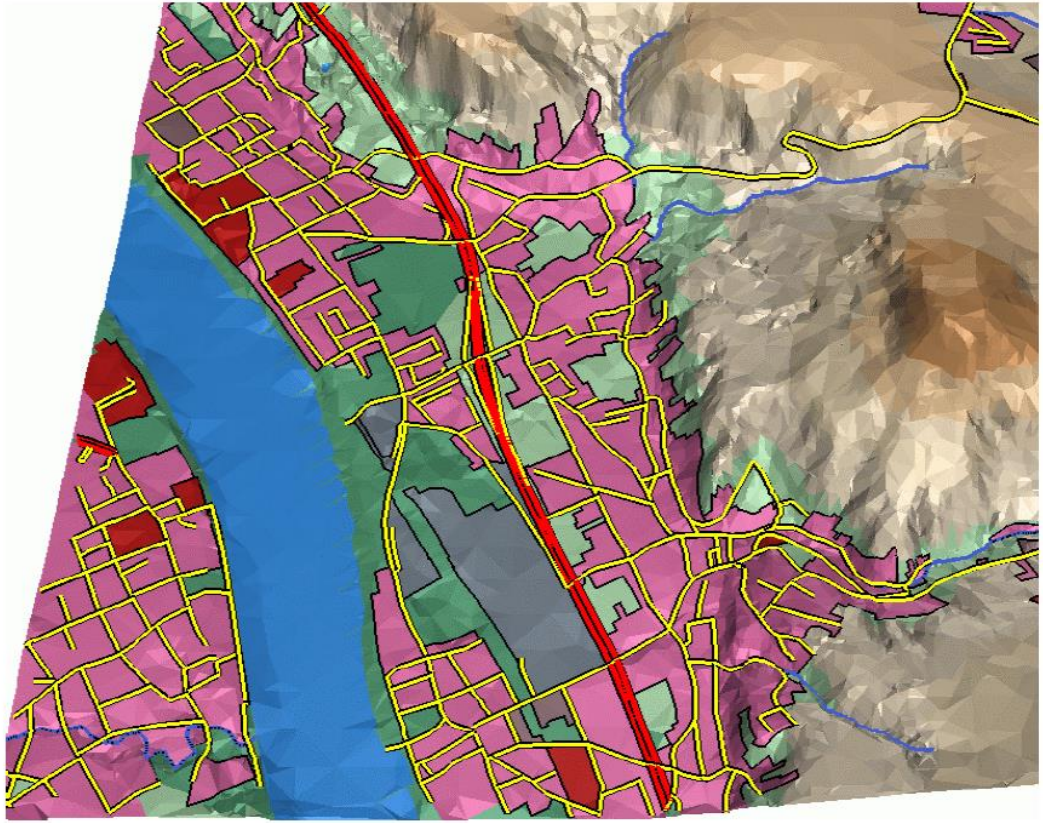


Fig. 72: LOD0 regional model consisting of land use objects in CityGML (source: IGG Uni Bonn).

Image source: CityGML 2.0 encoding standard specifications

CityGML: Waterbody model

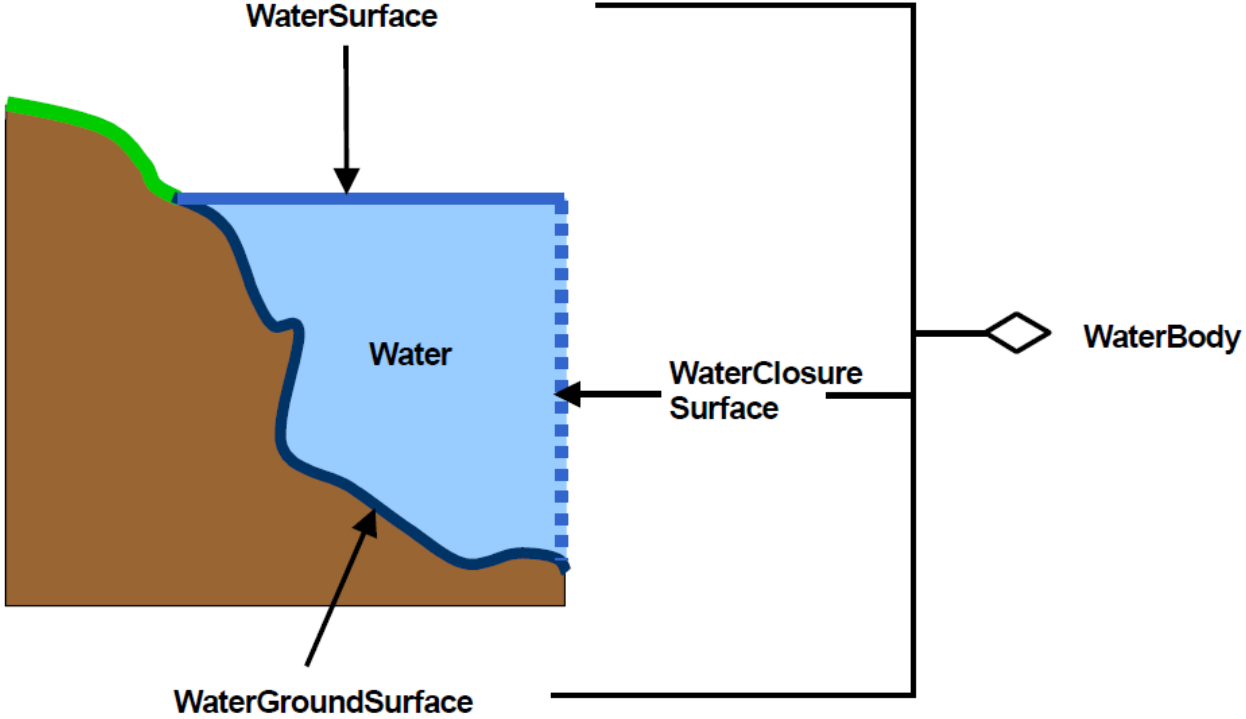


Fig. 55: Illustration of a water body defined in CityGML (graphic: IGG Uni Bonn).

CityGML: Terrain model

- Supports raster and vector DTMs
- Multiple, heterogeneous DTM can be nested
- Each DTM is delimited by a **validity extend polygon**

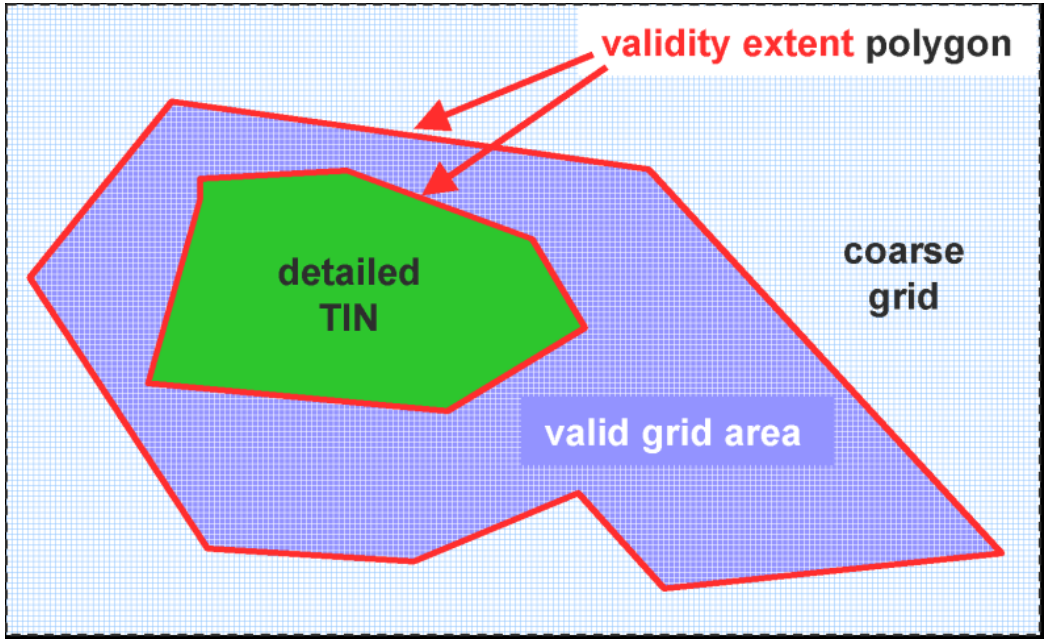


Image source: CityGML 2.0 encoding standard specifications

CityGML: Other modules

- **CityObjectGroup**
  - allows for arbitrary grouping of city objects
- **Generics**
  - allows to define generic city objects, which are not already defined
  - allows to define generic attributes, which are not already defined
- **Appearance**
  - allows to define one or multiple appearances for each city object
    - Styling with "colours"
    - Texturing

CityGML: beyond 3D geometry!

- 3D visualisation (geometry and graphical appearance) is just the very tip of the iceberg!

- CityGML objects have plenty of attributes, relations
  - They account for the core of semantic modelling
  - But, yes, these are less visible at a first sight…

Image source: https://colemaninsights.com/wp-content/uploads/2018/10/Iceberg.jpg

10th Int. CityGML Energy ADE workshop

A (less) gentle introduction to CityGML as open standard for semantic 3D city modelling

Giorgio Agugiaro

TU Delft, 5 December 2019

(City)GML: a closer look

- CityGML is based on GML
- GML (**Geography Markup Language**) serves as a standardised modelling language for geographic systems as well as an open interchange format for geographic transactions
- GML contains a set of ***primitive object types*** (think of basic types of Lego bricks…), such as:
  - Feature
  - Geometry
  - Coordinate reference system
  - Unit of measure
  - Etc.
- With GML, you can define your own object types for your application and create a specific ***domain application schema***. Those object types reference the primitives defined in the GML standard
- CityGML is therefore... an ***application schema*** based on GML

(City)GML: a closer look

- In GML, a major role is played by ***features***
  - A ***feature*** is an object representing a physical entity (e.g. a building, a river, or a person). A *feature* may or may not have geometric properties.
- A ***geometry object*** defines a location or region instead of a physical entity (and hence is different from a *feature)*.
- A *feature* can have various geometry properties (e.g.: a building can have different LoDs…)

- GML encodes geometries according to the "vector" model
- GML also allows features to **share a geometry property** with one another by using a *remote property reference* on the shared geometry property. An *xlink:href* attribute on a GML geometry property means that the value of the property is the resource referenced in the link

- Feeling lost? https://en.wikipedia.org/wiki/Geography_Markup_Language

CityGML: a closer look

- CityGML is actually two things
  a) It refers to the name of the **data model**
  b) It refers to one possible **encoding** of the data model

- The conceptual **data model** consists of UML diagrams (and the accompanying specifications)
- The **encoding** is how this information is actually written (e.g. to a file)
  – The most common encoding is by means of XML
  – The «rules» are encoded in a XSD file (XML Schema Definition)
  – The contents are written in a XML document «obeying» to the rules of the XSD file (the check is called «validation»)

- But there exist other encodings, e.g.
  – CityJSON (developed @ TU Delft)
  – SQL-based database model (3D City Database)

CityGML: a closer look

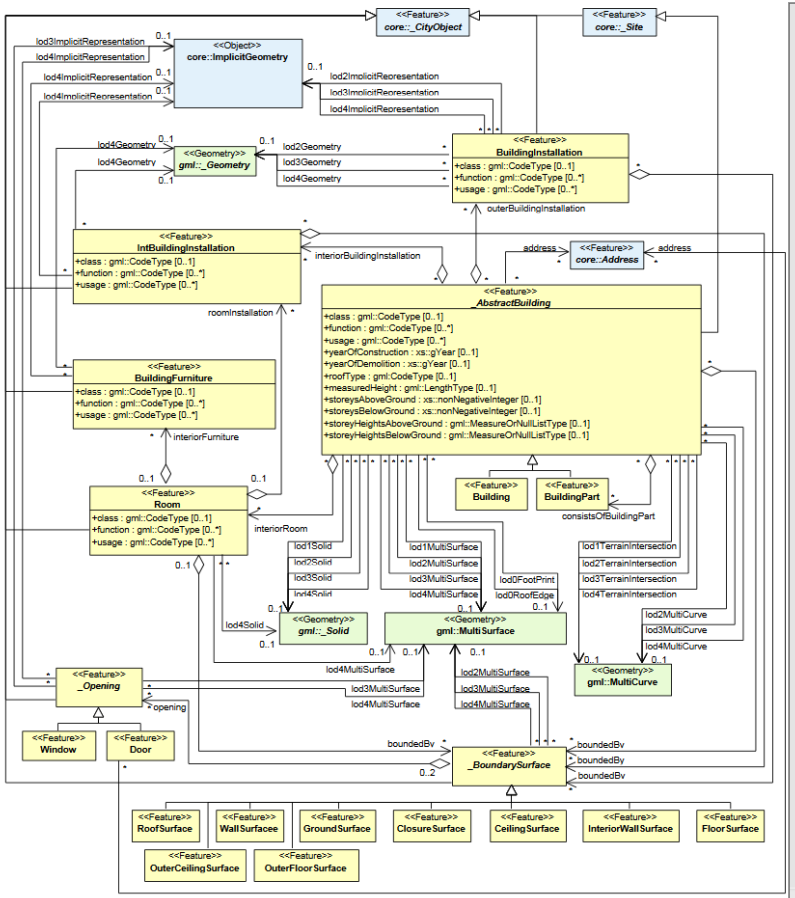
Semantic 3D city modelling

CityGML intro

UML in a nutshell

CityGML reprise

Conclusions



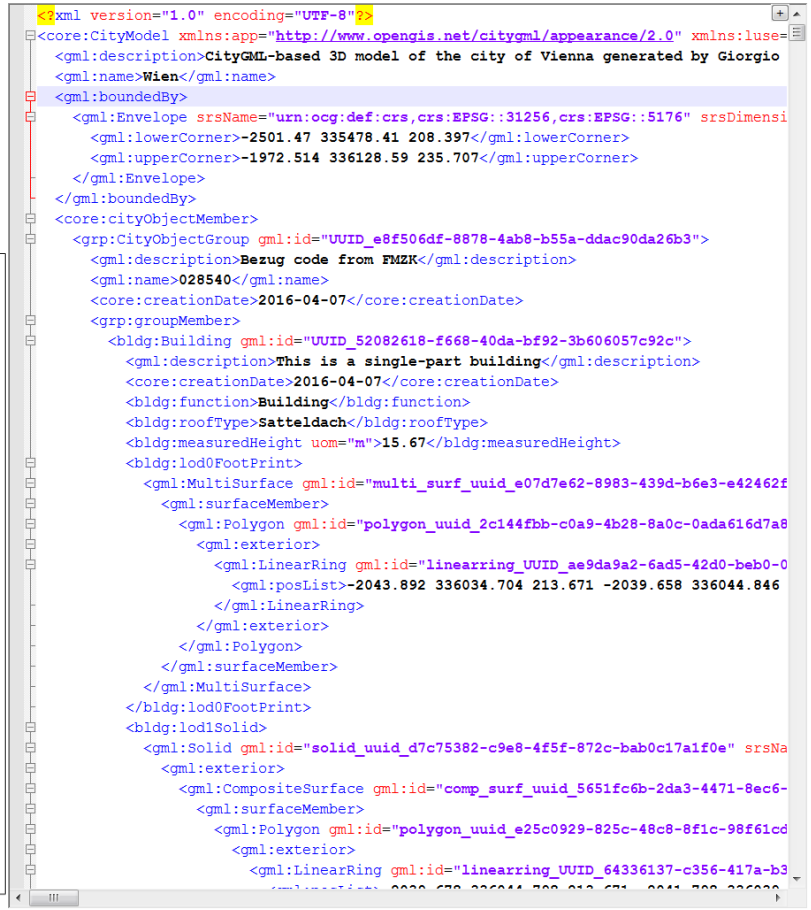


Image source: CityGML 2.0 encoding standard specifications

UML in a nutshell

UML is a general-purpose, development modelling language in the field of software engineering that is intended to provide a standard way to visualise the design of a system

When working with class diagrams, three are the main items

- **Classes** ("what")
  - Attributes
  - (Methods)
- **Multiplicity** ("how many")
- **Associations**

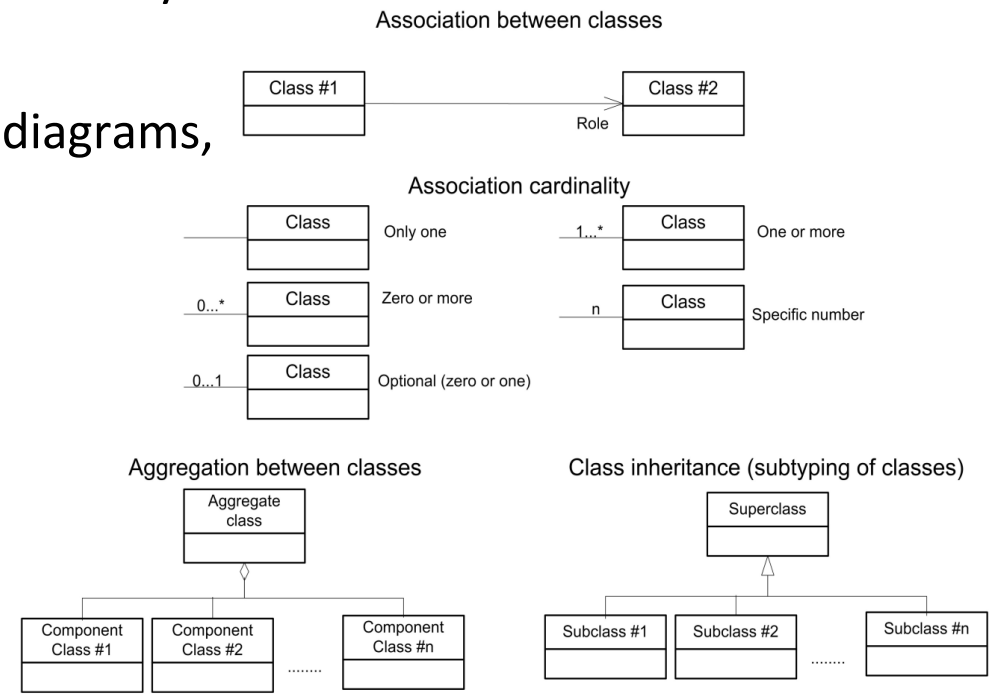


Image source: CityGML 2.0 encoding standard specifications

UML in a nutshell

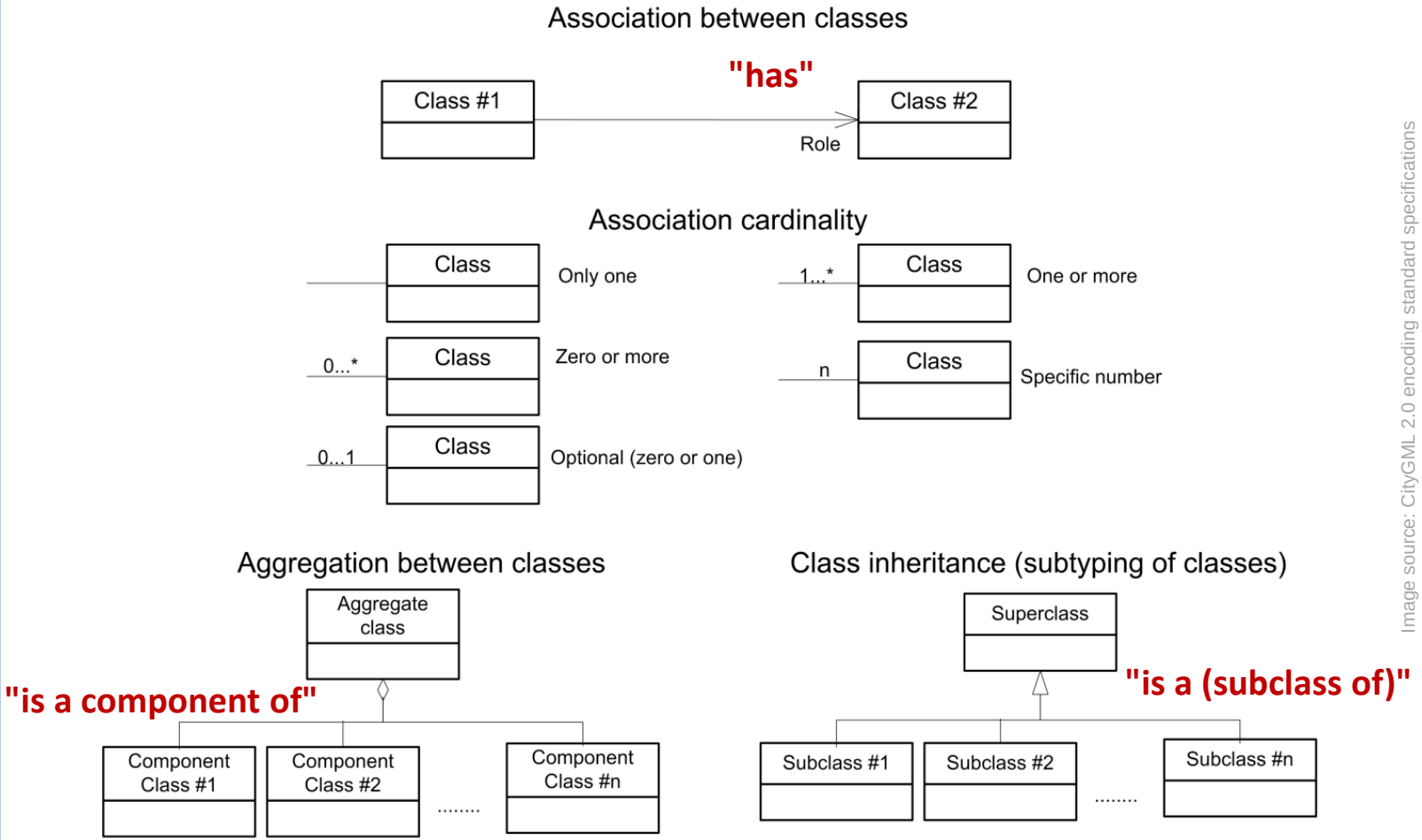
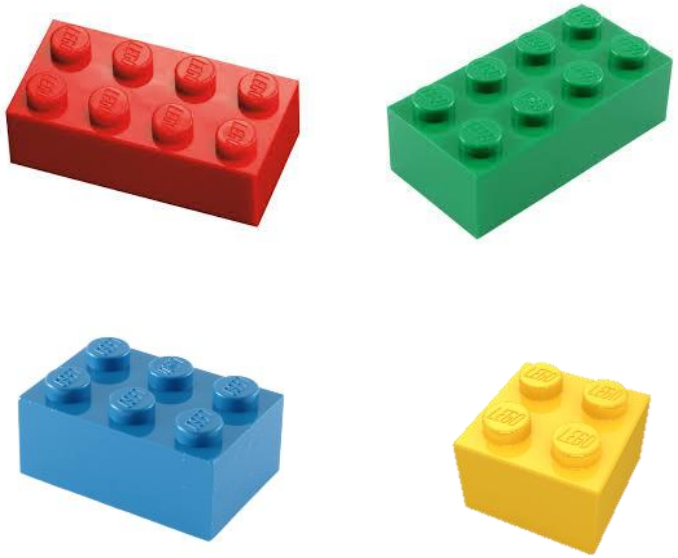


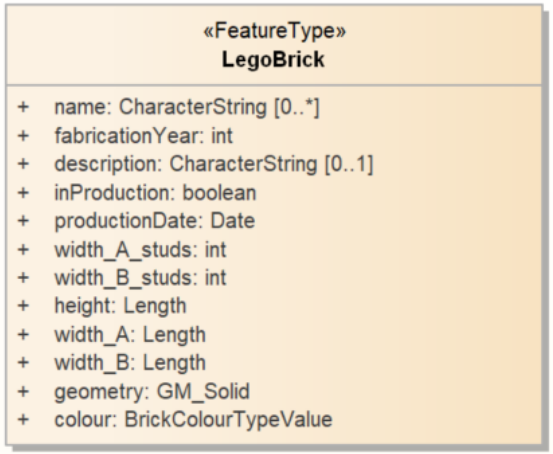
Image source: CityGML 2.0 encoding standard specifications

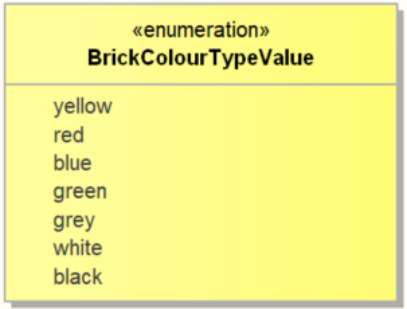
Image source: https://img.brickowl.com/files/image_cache/large/lego-green-brick-2-x-4-3001-30-771344-61.jpg



Objects
(Instances of the class)

Class

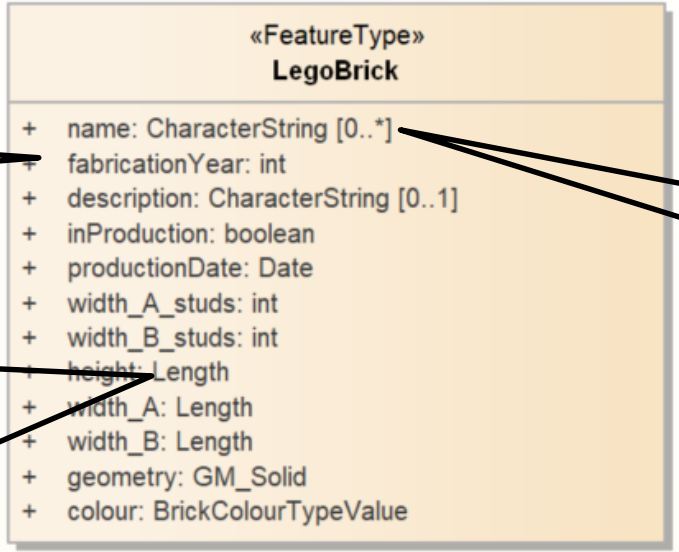




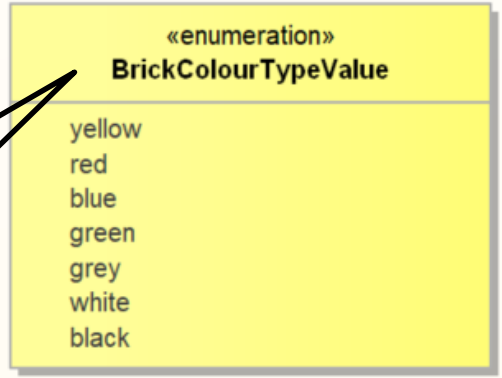
Attributes / Properties

Attributes / Properties **type**: Can have a simple or complex structure.
E.g. length = double (for the value) + CharacterString (for the UOM)

Enumerations contain *closed* sets of possible values
(**Codelists** contain *open* sets of possible values)



Multiplicity



Semantic 3D city modelling

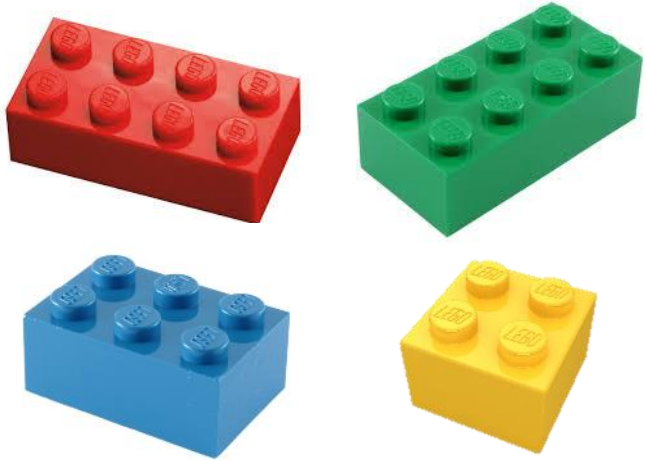
CityGML intro

UML in a nutshell

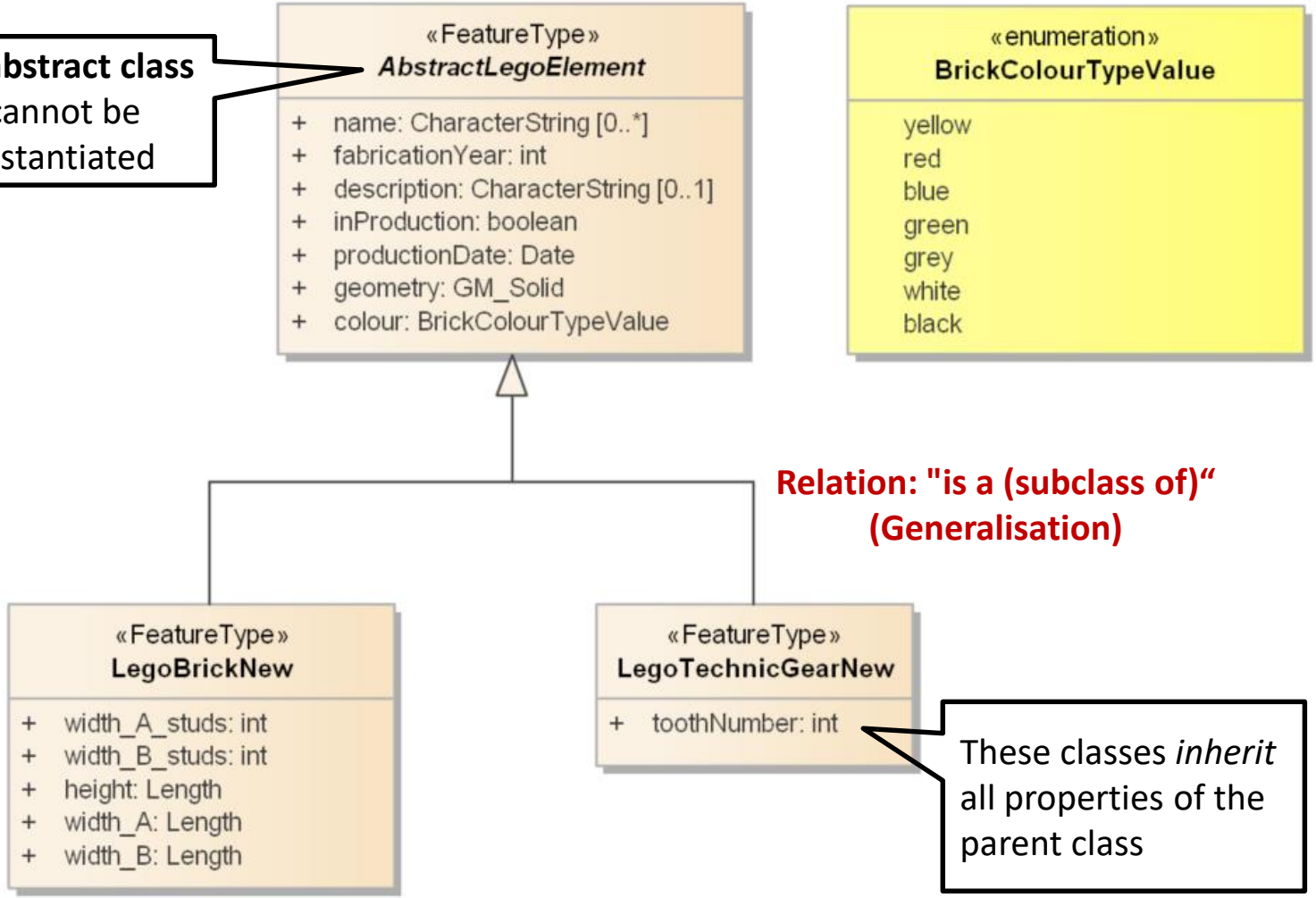
CityGML reprise

Conclusions

Image source: https://img.brickowl.com/files/image_cache/large/lego-green-brick-2-x-4-3001-30-771344-61.jpg



An **abstract class** cannot be instantiated



Relation: "is a (subclass of)" (Generalisation)

These classes *inherit* all properties of the parent class

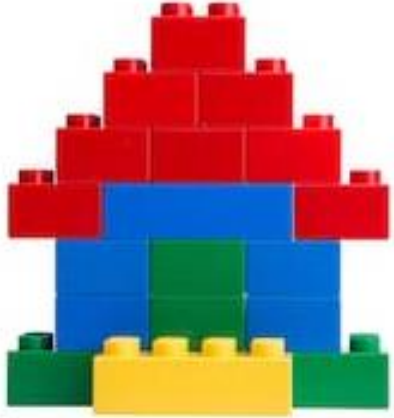
Semantic 3D city modelling

CityGML intro

UML in a nutshell

CityGML reprise

Conclusions



shutterstock.com • 300319532

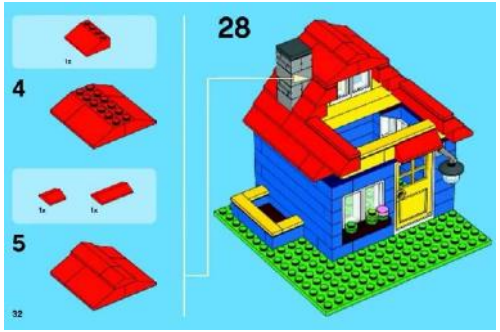


Image source: https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQR-9krd-BG7nipZE-eKorSihrNPTSDaVyCChoF6-Da11gmhdSdHw

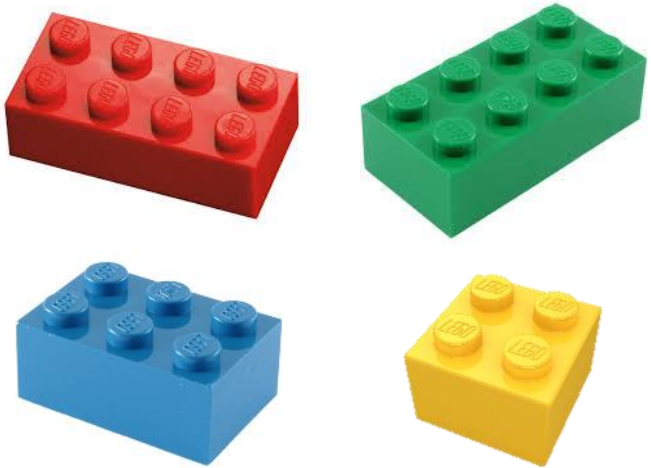
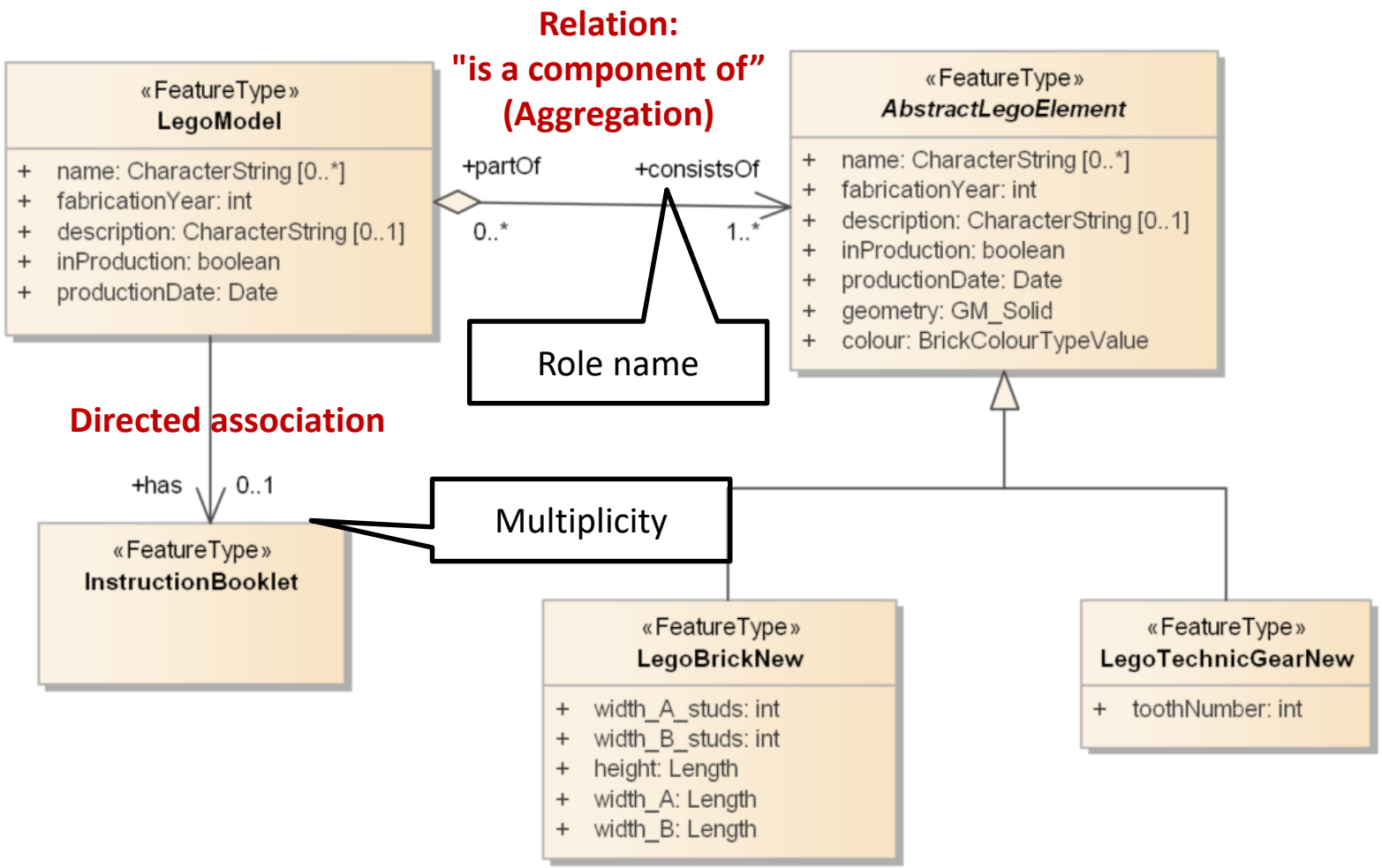


Image source: https://img.brickowl.com/files/image_cache/large/lego-green-brick-2-x-4-3001-30-771344-61.jpg



Semantic 3D city modelling

CityGML intro

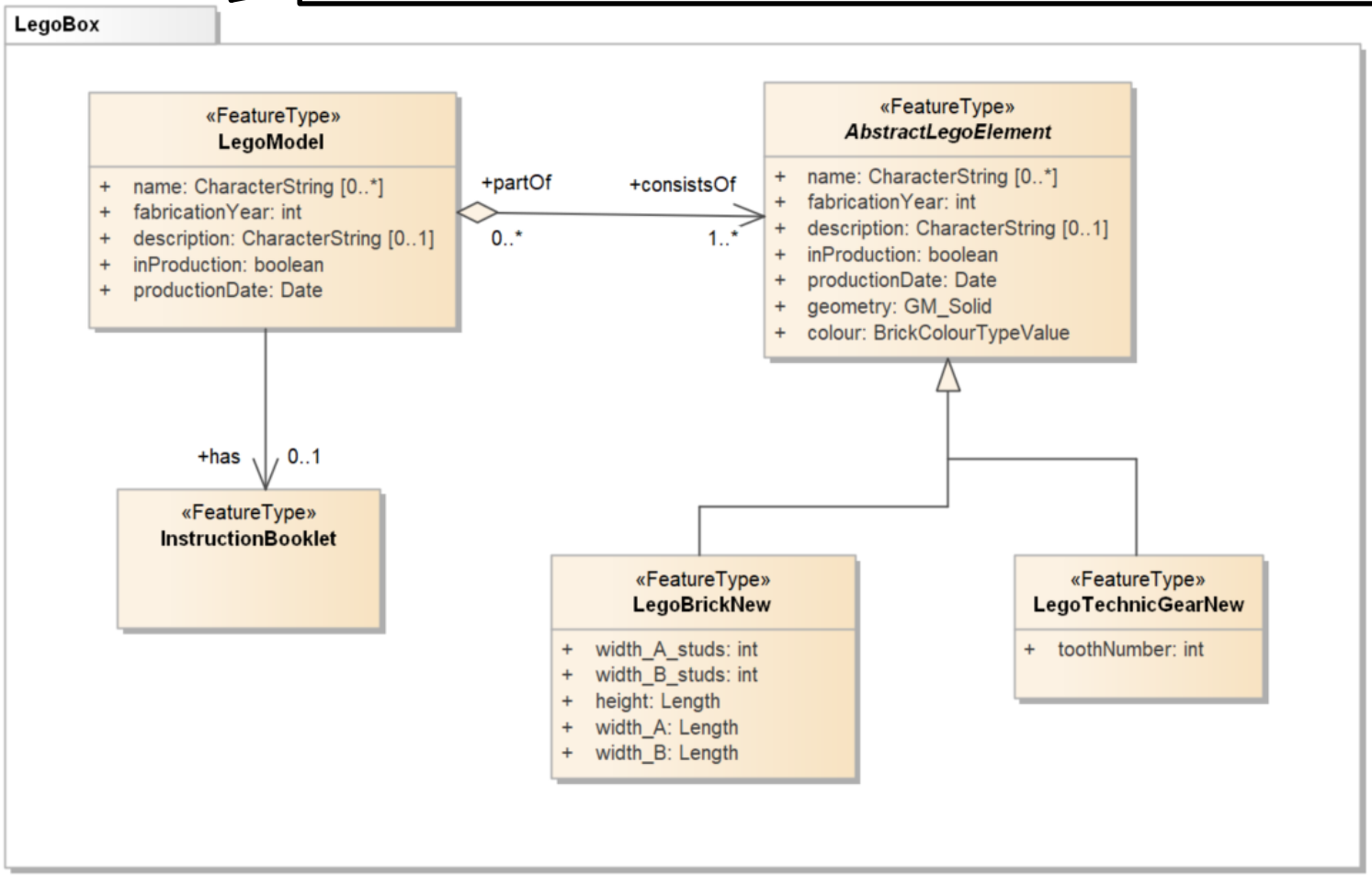
UML in a nutshell

CityGML reprise

Conclusions

Image source: https://www.lego.com

A package groups elements and provides a *namespace* for the grouped elements. A namespace is required to avoid name collisions



Back to CityGML

Semantic 3D city modelling

CityGML intro

UML in a nutshell

CityGML reprise

Conclusions

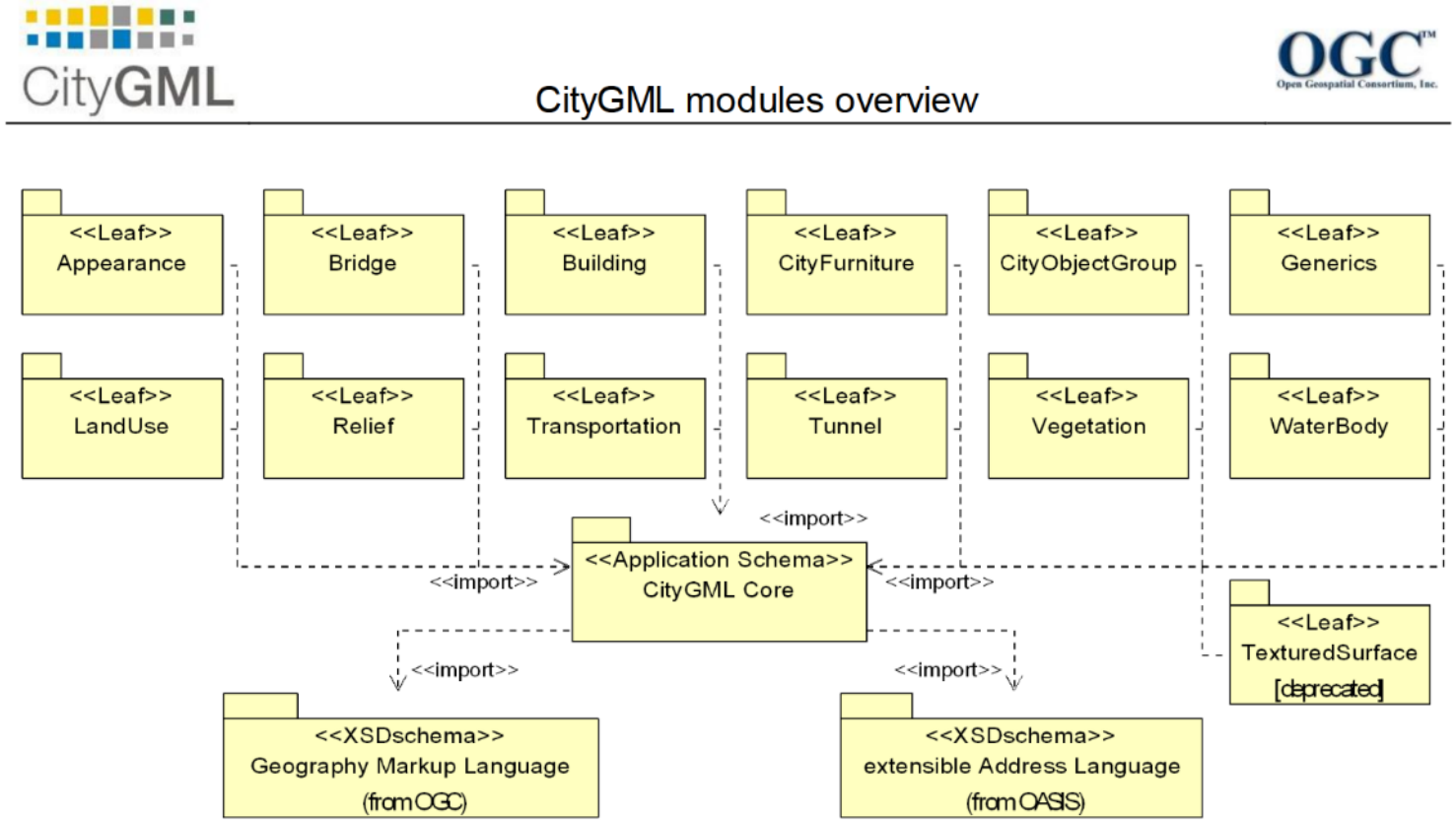


Image source: CityGML 2.0 encoding standard specifications

CityGML Core module, part 1

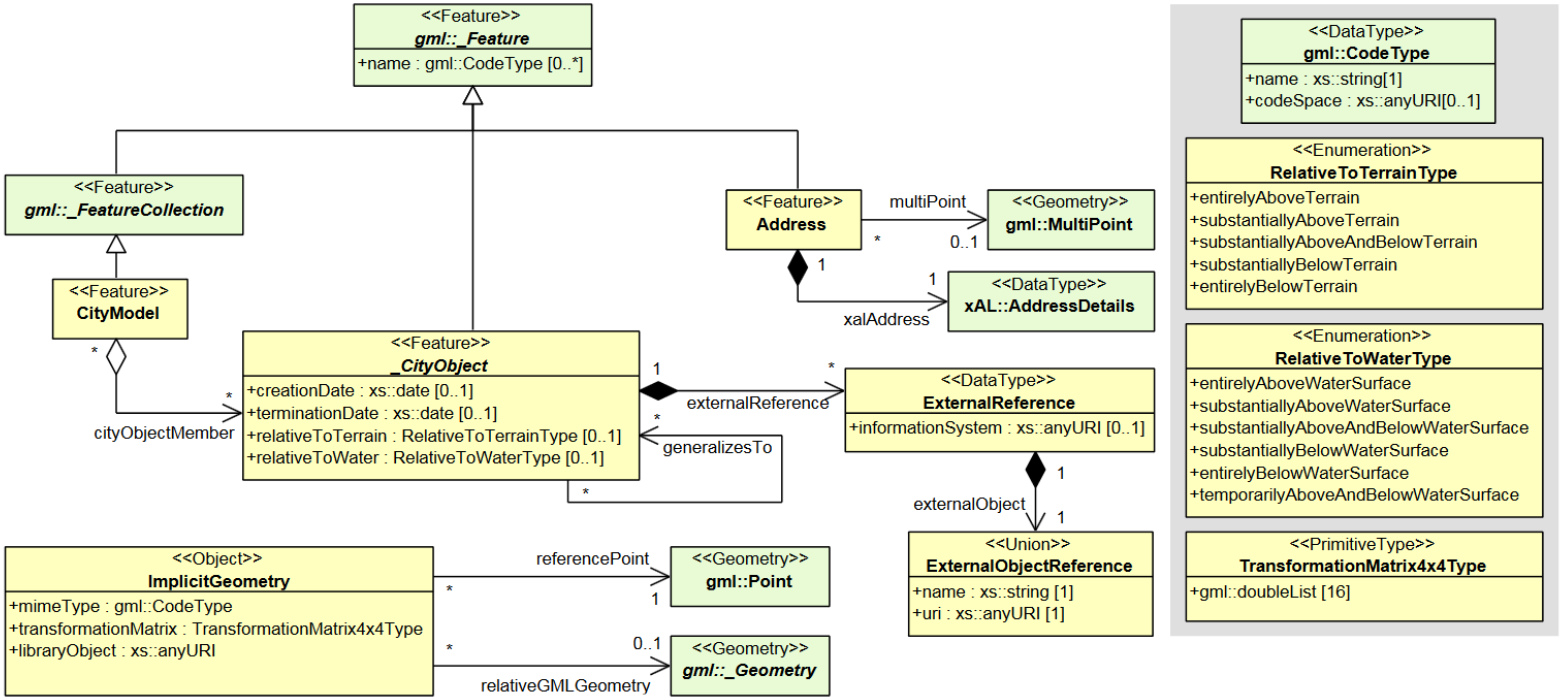


Image source: CityGML 2.0 encoding standard specifications

Semantic 3D city modelling

CityGML intro

UML in a nutshell

CityGML reprise

Conclusions

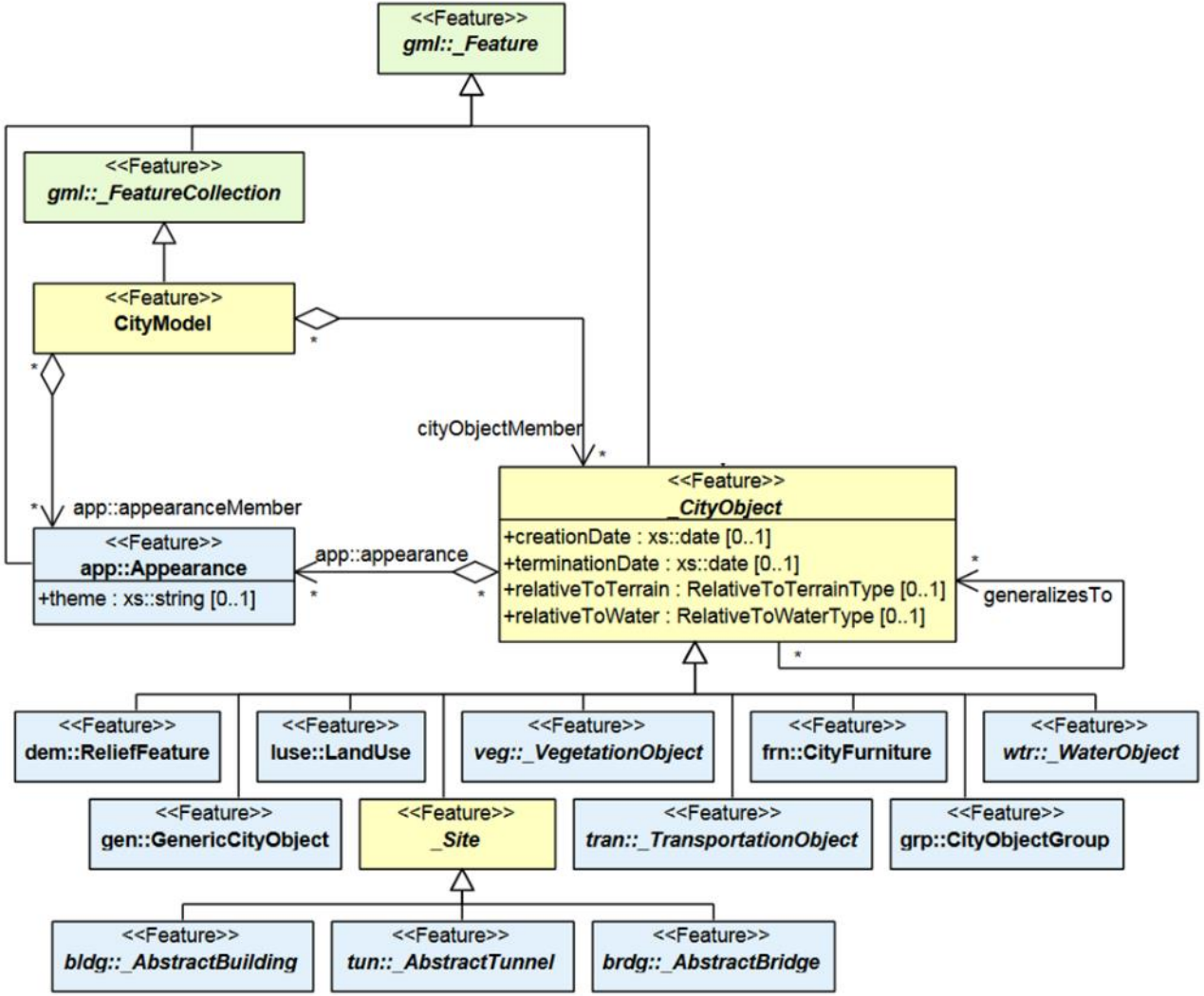


Image source: CityGML 2.0 encoding standard specifications

CityGML: vegetation model

- Solitary vegetation object can be represented in multiple LoDs with any geometry
- Plant cover can be represented only as MultiSurface or MultiSolid

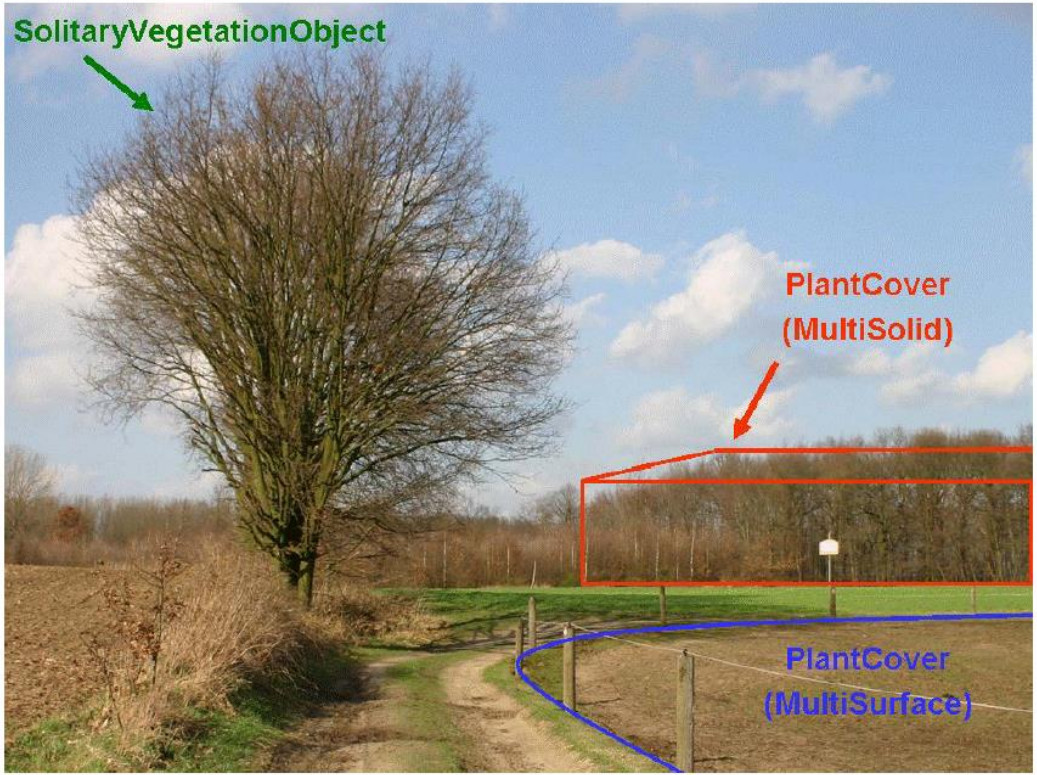


Fig. 63: Example for vegetation objects of the classes *SolitaryVegetationObject* and *PlantCover* (graphic: District of Recklinghausen).

CityGML: vegetation model

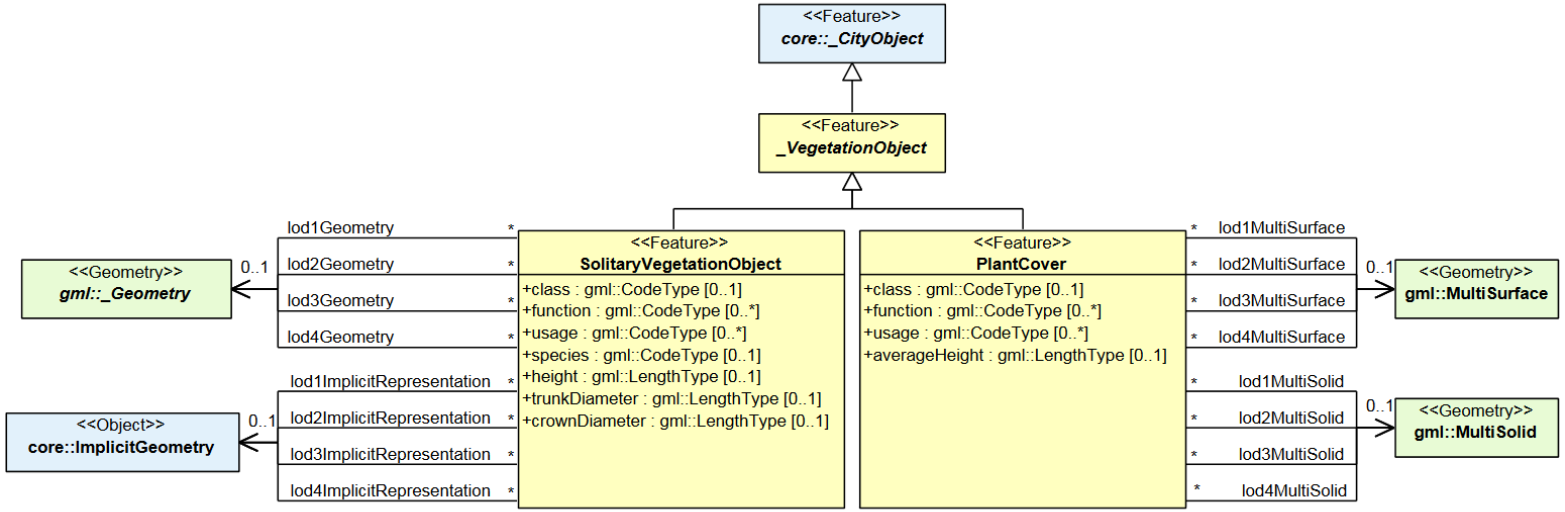


Image source: CityGML 2.0 encoding standard specifications

CityGML: vegetation model

Semantic 3D city modelling

CityGML intro

UML in a nutshell

CityGML reprise

Conclusions

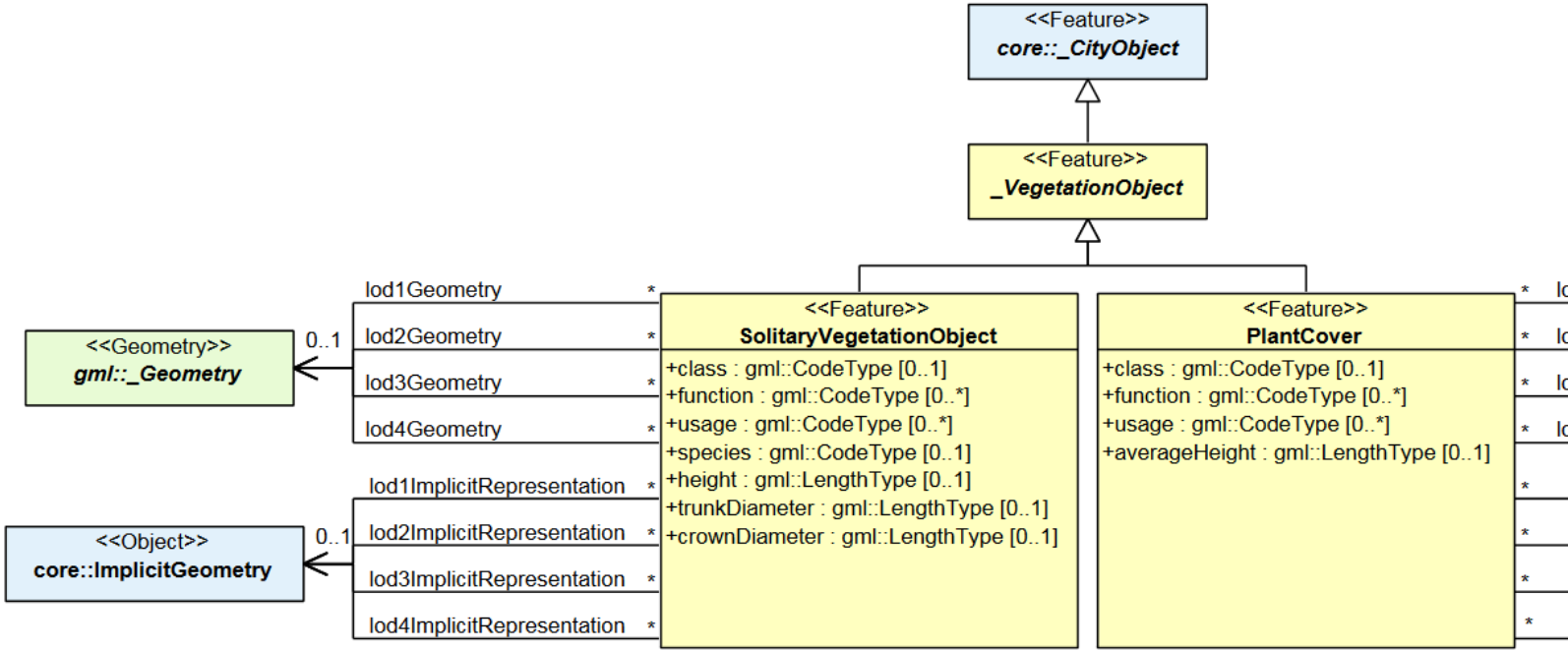


Image source: CityGML 2.0 encoding standard specifications

CityGML: building model

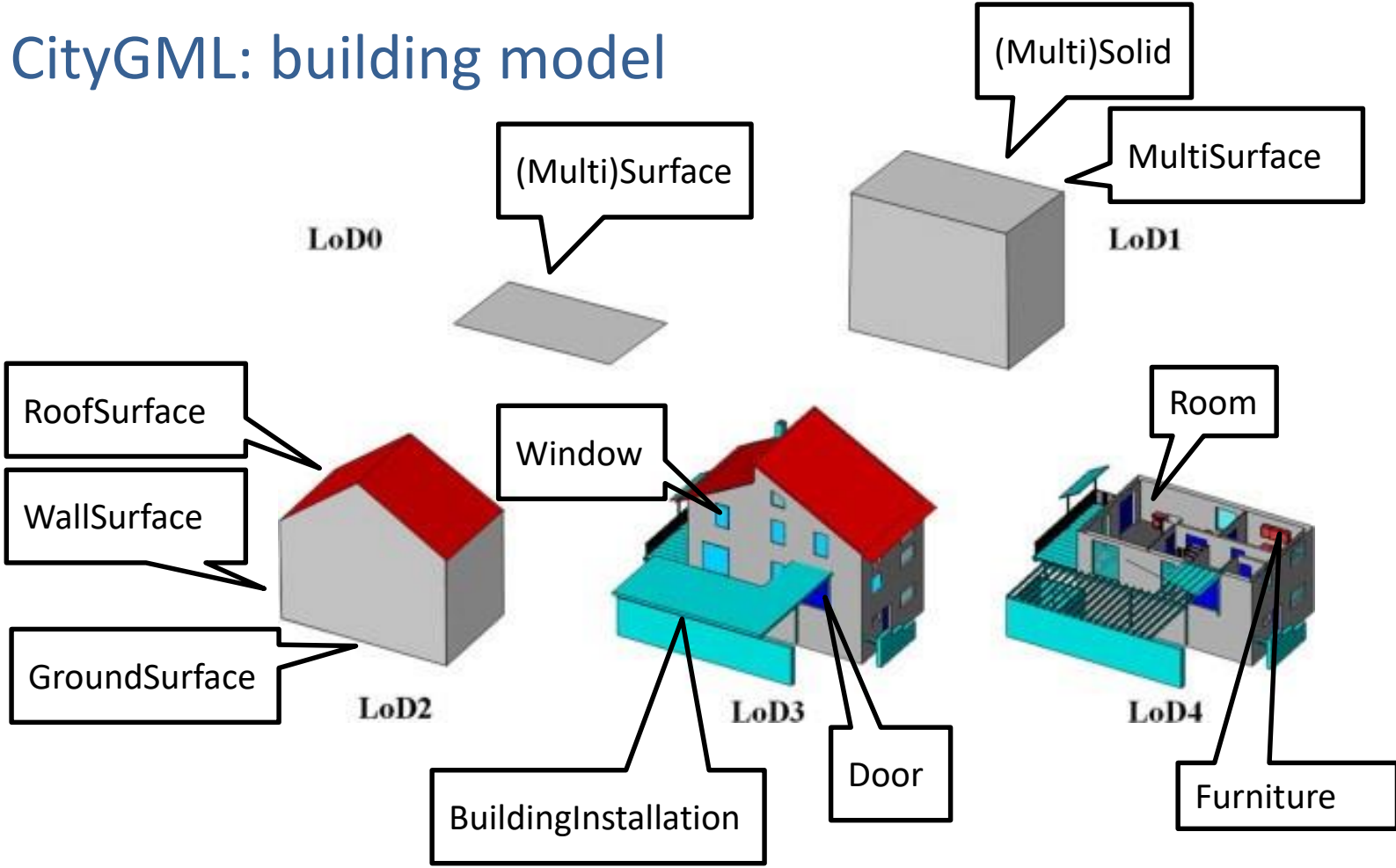


Image source: CityGML 2.0 encoding standard specifications

CityGML: building model

Image source: CityGML 2.0 encoding standard specifications

Semantic 3D city modelling

CityGML intro

UML in a nutshell

CityGML reprise

Conclusions

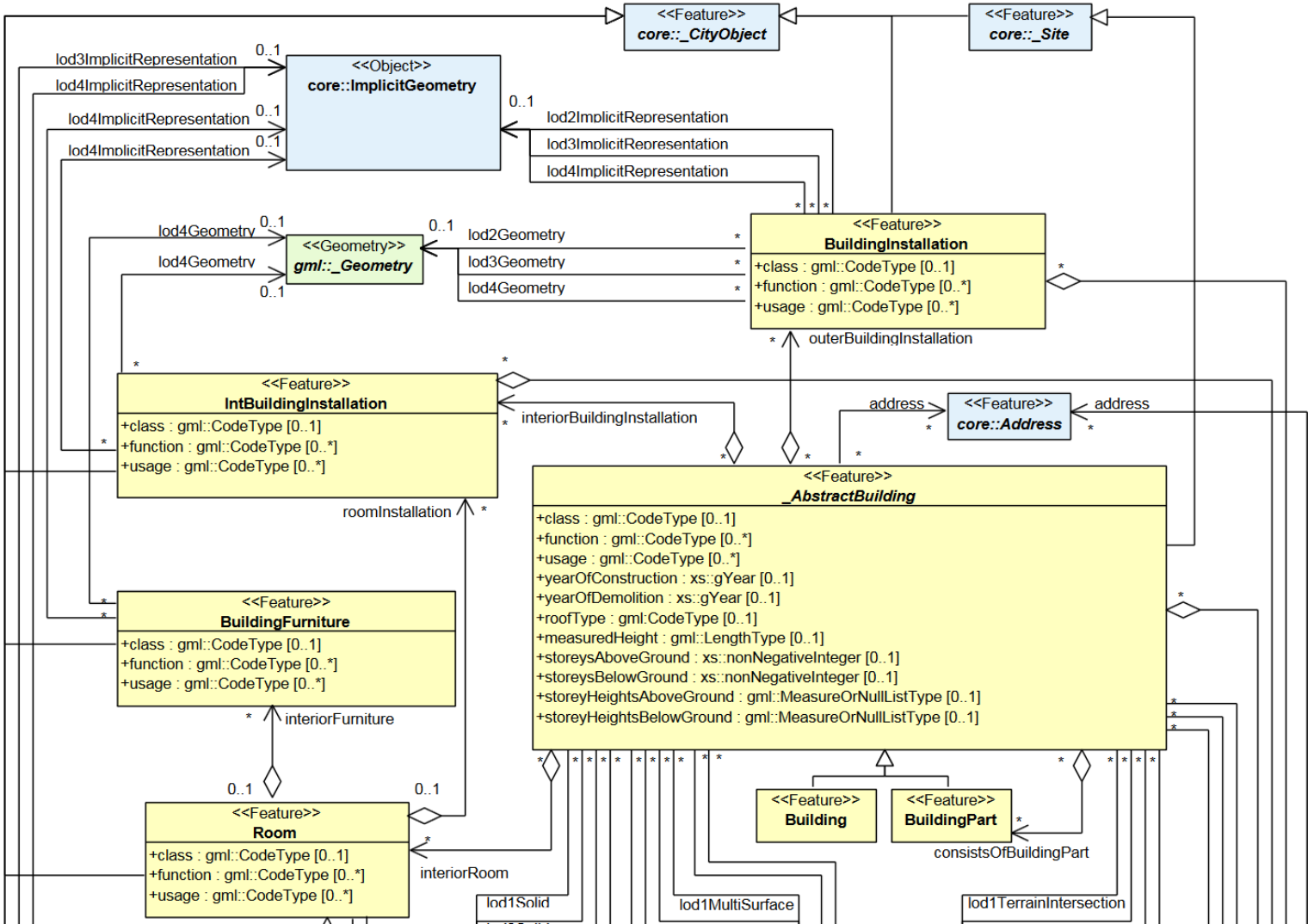


Image source: CityGML 2.0 encoding standard specifications

Semantic 3D city modelling

CityGML intro

UML in a nutshell

CityGML reprise

Conclusions

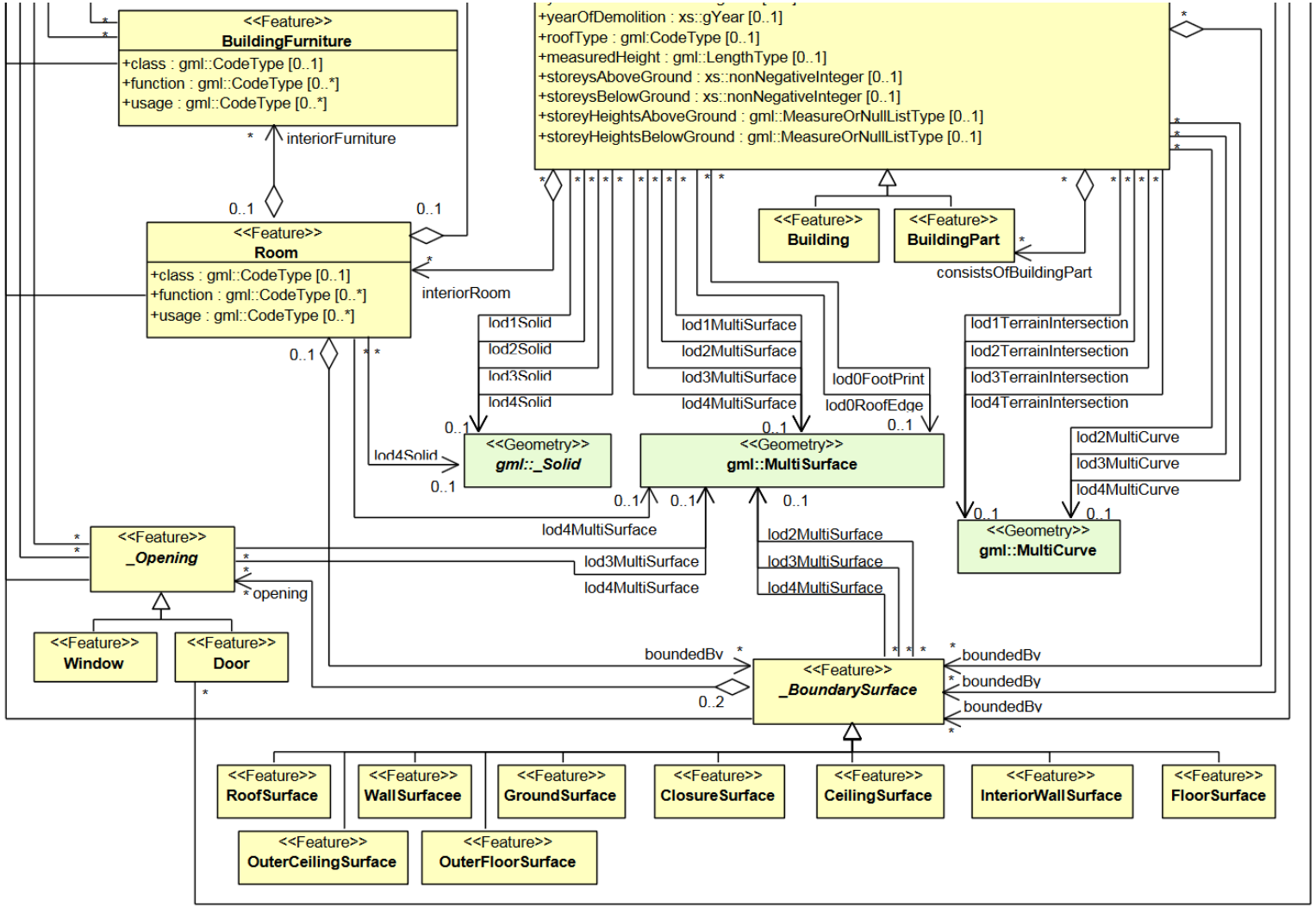


Image source: CityGML 2.0 encoding standard specifications

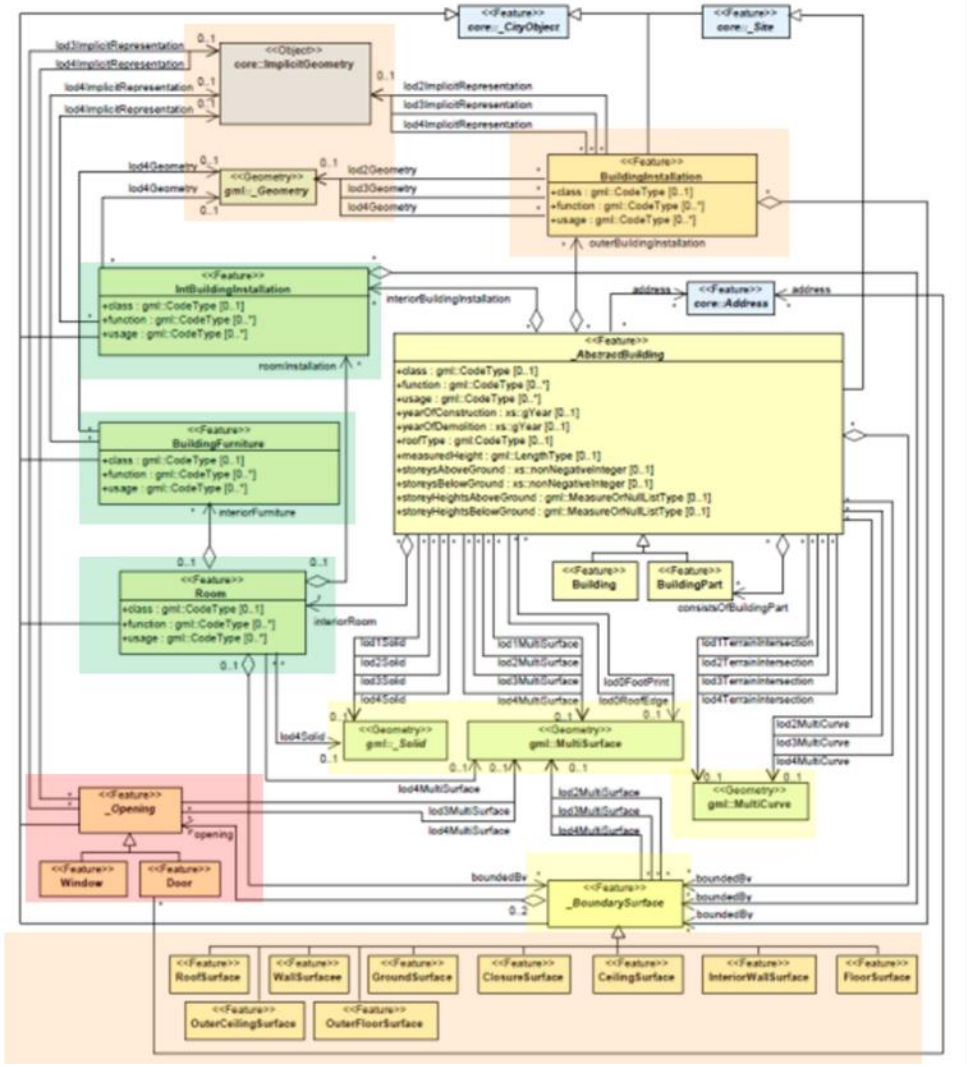
Semantic 3D city modelling

CityGML intro

UML in a nutshell

CityGML reprise

Conclusions



UML diagram of the building model in CityGML

Lod1

Lod2

Lod3

Lod4

Image source: CityGML 2.0 encoding standard specifications

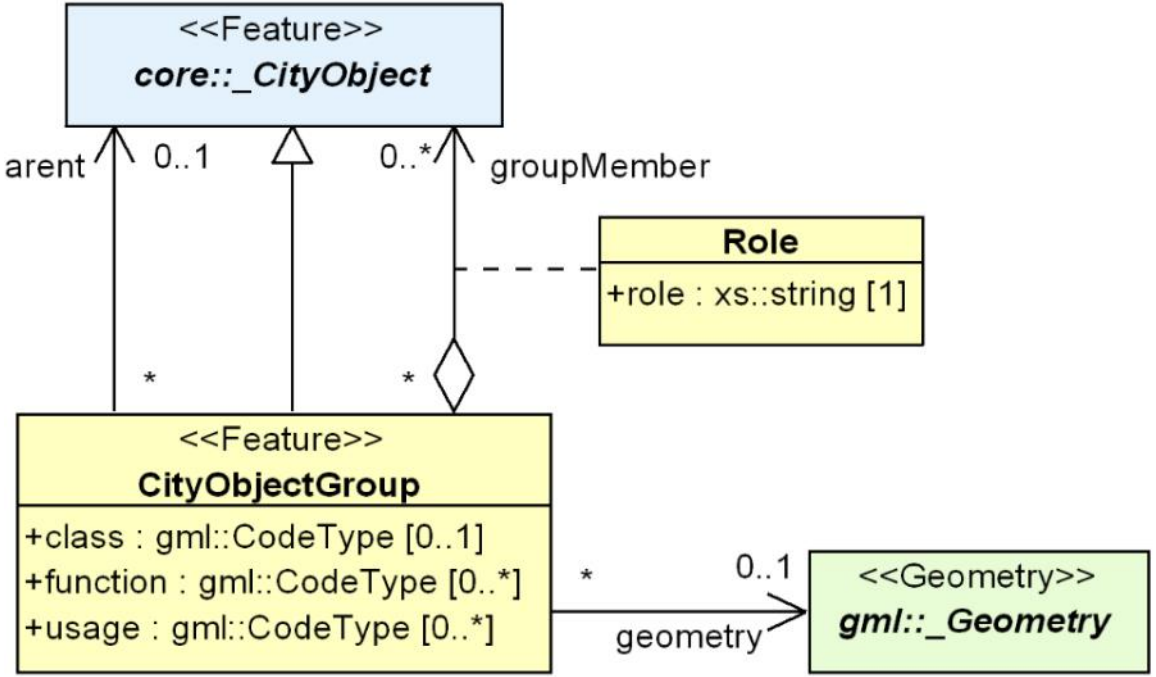
CityGML: Other modules

- **CityObjectGroup**
  - allows for arbitrary grouping of city objects

- **Generics**
  - allows to define generic city objects, which are not already defined
  - Allows to define generic attributes, which are not already defined

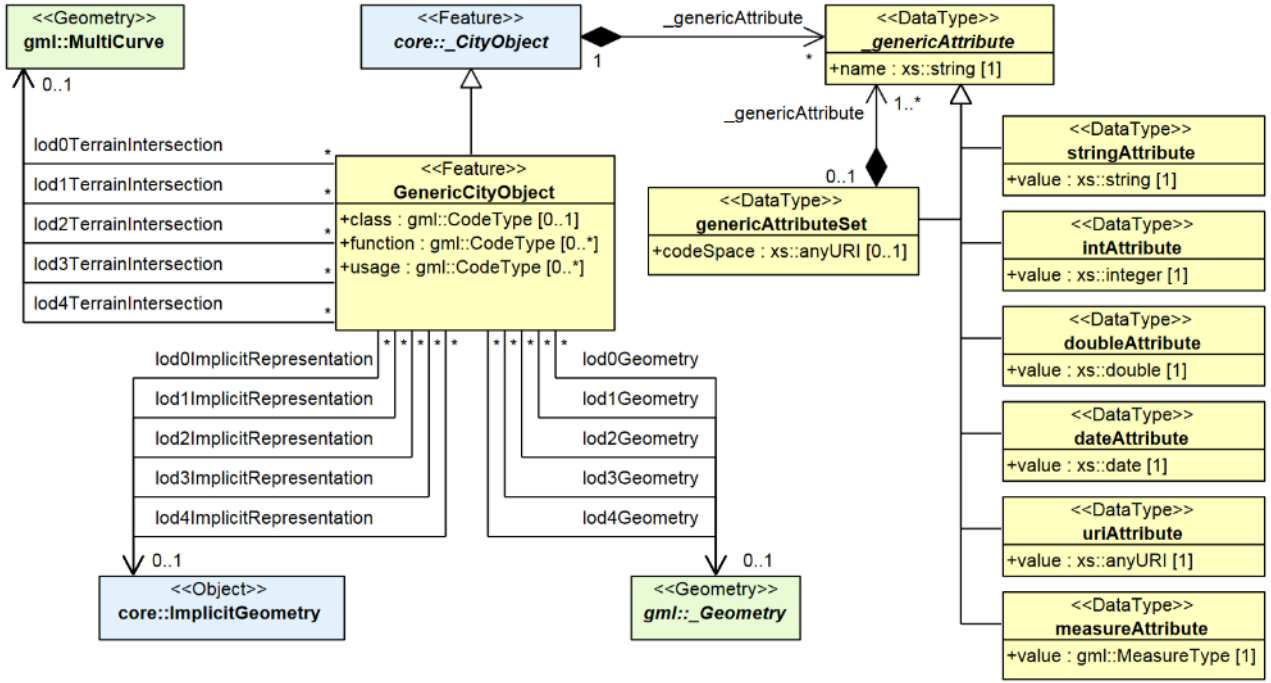
CityGML: CityObjectGroup model

- Allows for arbitrary (recursive) grouping of city objects



CityGML: generics model

- Allows to define **generic CityObjects** which are not already defined
- Allows to define **generic attributes** which are not already defined



Conclusions

- Feeling lost or overwhelmed? Do NOT worry, it is normal! ☺
- CityGML is an extremely vast and fascinating world, but…
- …it is also a dish that takes long to be appreciated (and digested!), *surely* longer than today's few hours during this crash course
- You do not necessarily need to focus on ALL "ingredients" (modules) at a time
- **You are not alone!**

Conclusions

Need help?

- CityGML homepage: http://www.citygml.org
- A more detailed course (by one of the "fathers" of CityGML): http://www.3dcitydb.org/3dcitydb/fileadmin/CityGML_Course/Course_CityGML.html
- RTFM: CityGML Encoding standard specifications http://www.opengeospatial.org/standards/citygml
- Last but not least: **Ask the experts** for assistance and cooperation, at TU Delft 3D Geoinfo …or elsewhere!

Thank you for your attention!

Dr. Giorgio Agugiaro
g.agugiaro@tudelft.nl
3D Geoinformation Group
TU Delft
The Netherlands
https://3d.bk.tudelft.nl/gagugiaro

TU Delft
3Dgeoinfo