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

Giorgio Agugiaro

TU Delft, 5 December 2019
License

This presentation is licensed under the Creative Commons License CC BY-NC-SA 3.0. According to CC BY-NC-SA 3.0 permission is granted to share this document, i.e. copy and redistribute the material in any medium or format, and to adapt it, i.e. remix, transform, and build upon the material under the following conditions:

• **Attribution:** You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

• **NonCommercial:** You may not use the material for commercial purposes.

• **ShareAlike:** If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

• **No additional restrictions:** You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.
City modelling

Real city

Digital twin

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

Processes

Actors

Entities

City model

represented by
City modelling: today

- Separate modelling, generally by specific sectors, e.g.
  - Energy
  - Mobility
  - Ecology
  - Economy
  - Surveying

Semantic 3D city modelling
CityGML intro
UML in a nutshell
CityGML reprise
Conclusions
Dealing with urban data...

- Semantic 3D city modelling
- CityGML intro
- UML in a nutshell
- CityGML reprise
- Conclusions
Which data model for cities?

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

Semantic 3D city modelling
CityGML intro
UML in a nutshell
CityGML reprise
Conclusions
What about standards?
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
Semantic 3D city modelling

CityGML intro

UML in a nutshell

CityGML reprise

Conclusions
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
CityGML: building model

Building with two building parts (represented as one Building feature and one included BuildingPart feature)

Building consisting of one part (represented as one Building feature)
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 implicit geometries
  - 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 City.Furniture objects in the right image (source: 3D city model of Borkenber).  

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

Semantic 3D city modelling

CityGML intro

UML in a nutshell

CityGML reprise

Conclusions
CityGML: land use model

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

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

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
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**
UML in a nutshell

**Association between classes**

"has"

```
Class #1

Role

Class #2
```

**Association cardinality**

- Only one: 1
- Zero or more: 0..*
- Optional (zero or one): 0..1
- One or more: 1..*
- Specific number: n

**Aggregation between classes**

"is a component of"

```
Aggregate class

Component Class #1

Component Class #2

....... Component Class #n
```

**Class inheritance (subtyping of classes)**

"is a (subclass of)"

```
Superclass

Subclass #1

Subclass #2

....... Subclass #n
```
Objects
(Instances of the 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)*
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
Semantic 3D city modelling

CityGML intro

UML in a nutshell

CityGML reprise

Conclusions

Relation: "is a component of" (Aggregation)

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

Multiplicity

Role name

Directed association

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

Image source: CityGML 2.0 encoding standard specifications
CityGML: building model

- RoofSurface
- WallSurface
- GroundSurface
- (Multi)Surface
- (Multi)Solid
- MultiSurface
- Window
- Door
- BuildingInstallation
- Room
- Furniture

Image source: CityGML 2.0 encoding standard specifications
CityGML: building model

[Image: Building with two building parts (represented as one Building feature and one included BuildingPart feature).]

[Image: Building consisting of one part (represented as one Building feature).]

Image source: CityGML 2.0 encoding standard specifications
Semantic 3D city modelling

CityGML intro

UML in a nutshell

CityGML reprise

Conclusions
Semantic 3D city modelling

CityGML intro

UML in a nutshell

CityGML reprise

Conclusions
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