CityGML Metadata ADE & links to Energy ADE

June 26, 2018
CityGML Energy ADE Workshop, Aachen

Kavisha
PhD Candidate
3D Geoinformation, TU Delft
Project Scope

• Metadata is essential for data discovery and fitness-for-purpose analysis

• There is currently very little support for metadata in CityGML

• Attribute-level metadata is already supported, e.g. the creationDate

• There is a need for metadata at the city model level and the city object/feature level
Approach

• Examined current geospatial metadata standards: selected ISO 19115 due to its wide range implementation and global approach (e.g. INSPIRE is too European-centric)

• Examined additional 3D geospatial needs through an extensive literature review and personal experience
Additional 3D elements

- Levels of detail (LOD), including room for the extended specifications of Biljecki et al. (2016a)
- Feature counts by LOD
- Presence of texture and/or material
- Datasets/Process used in acquisition and reconstruction (Lineage)
- Thematic Models included
- ADEs utilised and their metadata

CityGML Metadata ADE: Implementation
CityGML ADEs

Noise ADE

Energy ADE

Energy information

Any other application specific information
How it is modelled?
How it is modelled?
3D Metadata ADE Helper classes

- _MetadataModule
  - «featureType» _MetadataHelperClasses

- «featureType» ADEIdentifier
  + adeName: string
  + adeVersion: string
  + namespace: anyURI
  + status: string
  + authority: «Contact
  + summary: string [0..1]
  + urnSchema: anyURI [0..1]
  + urnModel: anyURI [0..1]
  + documentation: anyURI [0..1]

- «featureType» _Contact
  + contactName: string
  + phone: string
  + address: string
  + emailAddress: string

- «featureType» IndividualContact
  + role: MDroleCode [0..1]
  + organization: string [0..1]

- «featureType» OrganizationalContact
  + website: anyURI [0..1]

- «List» ConstraintsOnUsage
  + legalConstraints: MDlegalConstraints [0..*]
  + securityConstraints: MDsecurityConstraints [0..*]
  + userNote: string [0..1]

- «List» Lineage
  + source: string [0..1]
  + processStep: string [0..*]

- «List» presentThematicModels
  + thematicModel: ThematicModelCode [1..*]
Example Data: Outline

<md:MDcitymodel>
  <md:metadataIdentifier>MD_3D_MM_12345</md:metadataIdentifier>
  <md:citymodelIdentifier>MM_12345</md:citymodelIdentifier>
  <md:ISObased> [57 lines]
  <md:thematicModels> [14 lines]
  <md:textures>present</md:textures>
  <md:materials>present</md:materials>
  <md:ADEmetadata> [20 lines]
  <md:MDcityfeatures> [13 lines]
  <md:MDcityfeatures> [14 lines]
  <md:MDcityfeatures> [13 lines]
  <md:MDcityfeatures> [13 lines]
  <md:MDcityfeatures> [13 lines]
  <md:MDcityfeatures> [15 lines]
  <md:MDcityfeatures> [11 lines]
  <md:MDcityfeatures> [11 lines]
  <md:MDcityfeatures> [11 lines]
  <md:MDcityfeatures> [11 lines]
  <md:MDcityfeatures> [11 lines]
  <md:MDcityfeatures> [11 lines]
  <md:MDcityfeatures> [11 lines]
  <md:MDcityfeatures> [17 lines]
  <md:MDcityfeatures> [23 lines]
  <md:MDcityfeatures> [17 lines]
  <md:MDcityfeatures> [17 lines]
  <md:LevelsOfDetail> [5 lines]
  <md:LevelsOfDetail> [5 lines]
</md:MDcitymodel>
<md:ISOmetadata>
  
  <md:ISOIdentifier>
    <md:datasetTitle>3D City Model</md:datasetTitle>
    <md:datasetReferenceDate>2017-01-01</md:datasetReferenceDate>
    <md:geoLocation>Delft</md:geoLocation>
    <md:datasetLanguage>English</md:datasetLanguage>
    <md:datasetTopicCategory>Metadata associated with the 3D city model</md:datasetTopicCategory>
    <md:datasetDescription>Metadata associated with the 3D city model</md:datasetDescription>
    <md:distributionFormatVersion>CityGML 2.0</md:distributionFormatVersion>
    <md:spatialRepresentationType>CityGML 2.0</md:spatialRepresentationType>
    <md:temporalInformation>2017-01-02</md:temporalInformation>
    <md:onlineResource>https://3d.bk.tudelft.nl/opendata/3dfier/</md:onlineResource>
    <md:fileIdentifier>Sample01.xml</md:fileIdentifier>
    <md:metadataStandard>ISO 19115</md:metadataStandard>
    <md:metadataStandardVersion>v01</md:metadataStandardVersion>
    <md:metadataLanguage>English</md:metadataLanguage>
    <md:metadataCharacterSet>UTF-8</md:metadataCharacterSet>
    <md:metadataPointOfContact>2017-02-01</md:metadataPointOfContact>
    <md:metadataDateStamp>2017-02-01</md:metadataDateStamp>
    <md:lineage>5 lines</md:lineage>
    <md:boundingBox3D>5 lines</md:boundingBox3D>
    <md:abstract>Metadata associated with the 3D city model</md:abstract>
    <md:specificUsage>Geoinformation model</md:specificUsage>
    <md:keywords>3D CityGML, 3D Metadata</md:keywords>
    <md:constraints>6 lines</md:constraints>
  </md:ISOIdentifier>
</md:ISOmetadata>
Example Data: Thematic models

<md:thematicModels>
  <md:presentThematicModels>
    <md:thematicModel>Building</md:thematicModel>
    <md:thematicModel>Bridge</md:thematicModel>
    <md:thematicModel>Tunnel</md:thematicModel>
    <md:thematicModel>Vegetation</md:thematicModel>
    <md:thematicModel>WaterBody</md:thematicModel>
    <md:thematicModel>Transportation</md:thematicModel>
    <md:thematicModel>Generics</md:thematicModel>
    <md:thematicModel>CityFurniture</md:thematicModel>
    <md:thematicModel>CityObjectGroup</md:thematicModel>
    <md:thematicModel>Relief</md:thematicModel>
  </md:presentThematicModels>
</md:thematicModels>
Example Data: ADE metadata

<md:ADEmetadata>
  <md:ADEidentifier>
    <md:adeName>CityGML iTINs ADE</md:adeName>
    <md:adeVersion>0.1</md:adeVersion>
    <md:namespace>http://godzilla.bk.tudelft.nl/schemas/CityGML_iTINs_ADE</md:namespace>
    <md:status>Implemented</md:status>
    <md:authority>
      <md:OrganizationalContact>
        <md:contactName>3D Geoinformation Group</md:contactName>
        <md:phone>+31 666666666</md:phone>
        <md:address>Julianalaan 134, 2628 BL, TU Delft</md:address>
        <md:emailAddress>3dtud@tudelft.nl</md:emailAddress>
        <md:website>https://3d.bk.tudelft.nl</md:website>
      </md:OrganizationalContact>
    </md:authority>
    <md:summary>CityGML ADE for storing massive TIN terrains.</md:summary>
    <md:xmlSchema>https://github.com/tudelft3d/CityGML_iTINs_ADE/blob/master/XSD/CityGML_iTINs_ADE.xsd</md:xmlSchema>
    <md:umlModel>https://github.com/tudelft3d/CityGML_iTINs_ADE/blob/master/UML/CityGML_iTINs_ADE_0_1.eap</md:umlModel>
    <md:documentation>https://github.com/tudelft3d/CityGML_iTINs_ADE/tree/master/Documentation</md:documentation>
  </md:ADEidentifier>
</md:ADEmetadata>
Example Data: City features

```xml
<md:MDcityfeatures>
    <md:MDbuilding>
        <md:featureType>Building</md:featureType>
        <md:featureCount>100</md:featureCount>
        <md:LevelsOfDetail>
            <md:LevelOfDetail>
                <md:lod>1</md:lod>
                <md:objectCount>100</md:objectCount>
            </md:LevelOfDetail>
        </md:LevelsOfDetail>
    </md:MDbuilding>
</md:MDcityfeatures>
```
3D Metadata ADE Codelist & Enumerations

class 3DMD_Codelists

- farming
- biota
- boundaries
- climatology
- meteorology
- atmosphere
- economy
- elevation
- environment
- geoscientificInformation
- health
- imagery
- BaseMaps
- EarthCover
- intelligence
- Military
- inlandWaters
- location
- oceans
- planningCadastre
- society
- structure
- transportation
- utilities
- Communication
- extraTerrestrial
- disaster

class 3DMD_Codelist

<codeList> MDTopicCategory

+ farming
+ biota
+ boundaries
+ climatology
+ meteorology
+ atmosphere
+ economy
+ elevation
+ environment
+ geoscientificInformation
+ health
+ imagery
+ BaseMaps
+ EarthCover
+ intelligence
+ Military
+ inlandWaters
+ location
+ oceans
+ planningCadastre
+ society
+ structure
+ transportation
+ utilities
+ Communication
+ extraTerrestrial
+ disaster

<codeList> MDRoleCode

+ resourceProvider
+ custodian
+ owner
+ user
+ distributor
+ originator
+ pointOfContact
+ principalInvestigator
+ processor
+ publisher
+ author
+ sponsor
+ co-author
+ collaborator
+ editor
+ mediator
+ rightsHolder
+ contributor
+ funder
+ stakeholder

<codeList> MDlegalConstraints

+ copyright
+ patent
+ patentPending
+ trademark
+ licence
+ intellectualPropertyRights
+ restricted
+ otherRestrictions
+ unstructured
+ licenseUnrestricted
+ licenseEndUser
+ licenseDistributor
+ private
+ statutory
+ confidential
+ sensitiveButUnclassified
+ in-confidence

<codeList> MDsecurityConstraints

+ unclassified
+ restricted
+ confidential
+ secret
+ topSecret
+ sensitiveButUnclassified
+ forOfficialUseOnly
+ protected
+ limitedDistribution

<codeList> MDspatialRepTypeCode

+ Vector
+ grid
+ TIN
+ testTable
+ stereoModel
+ video

<codeList> TerrainTypeCode

+ Attributes
+ TINRelief
+ RasterRelief
+ MassPointRelief
+ BreaklineRelief

<enumeration> LODcode

<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>0.0</td>
</tr>
<tr>
<td>0.1</td>
</tr>
<tr>
<td>0.2</td>
</tr>
<tr>
<td>0.3</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1.0</td>
</tr>
<tr>
<td>1.1</td>
</tr>
<tr>
<td>1.2</td>
</tr>
<tr>
<td>1.3</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>2.0</td>
</tr>
<tr>
<td>2.1</td>
</tr>
<tr>
<td>2.2</td>
</tr>
<tr>
<td>2.3</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>3.0</td>
</tr>
<tr>
<td>3.1</td>
</tr>
<tr>
<td>3.2</td>
</tr>
<tr>
<td>3.3</td>
</tr>
<tr>
<td>4</td>
</tr>
</tbody>
</table>

<enumeration> StateCode

<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>present</td>
</tr>
<tr>
<td>absent</td>
</tr>
</tbody>
</table>
Extending the 3D Metadata ADE

• The ADE is modularly designed for future possible extensions to store metadata related to other domains and applications.

• It can be extended by other ADEs to incorporate domain-specific data needs, e.g. the Energy ADE (number of thermal zones, building units, etc.)
CityGML 3D Metadata ADE

CityGML extension for modelling metadata related to 3D city models. This ADE contains ISO 19115 compliant elements derived from the Metadata for the discovery of geographic datasets and series table (Table F.1 in the ISO 19115:2014 documentation). This ADE also contains 3D specific information such as the levels of details, thematic models, semantics, textures, materials and process of model acquisition.

Resources

The UML model of the 3D Metadata ADE was created using Enterprise Architect.

* This [Enterprise Architect file](#) contains the UML model.
* The UML diagrams can be downloaded in [PDF format](#) as well.

The [XML Schema file and codetlists](#) of the 3D Metadata ADE were derived automatically from the UML model using...
Automatic metadata creation for current datasets using Python

******* Metadata Generation *******

CityGML input file:  citygmldatasets/Part-2-Tunnel-Bridge-V3.gml

Metadata output file:  citygmldatasets/Part-2-Tunnel-Bridge-V3_metadata.gml

Reading elements ......

citymodelIdentifier not defined!! An arbitrary value is given.

ISO metadata:
datasetTitle is not defined!!
datasetReferenceDate is not defined!!
datasetResponsibleParty is not defined!!
geoLocation is not defined!!
datasetDescription is not defined!!
temporalInformation is not defined!!
onlineResource is not defined!!
datasetResponsibleParty is not defined!!
lineage source and process steps are not defined!!
abstract is not defined!!
specificUsage is not defined!!
keywords is not defined!!
constraints not defined!!

Thematic Models Present:
Tunnel
Bridge

Time taken for metadata generation:  0.1049652099609375 sec
CityJSON is a format for encoding a subset of the CityGML data model using JavaScript Object Notation (JSON). A CityJSON file represents both the geometry and the semantics of the city features of a given area, eg buildings, roads, rivers, the vegetation, and the city furniture.

The aim of CityJSON is to offer an alternative to the GML encoding of CityGML, which can be verbose and complex (and thus rather frustrating to work with). CityJSON aims at being easy-to-use, both for reading datasets, and for creating them. It was designed with programmers in mind, so that tools and APIs supporting it can be quickly built. It
Metadata in CityJSON

- CityJSON v0.7 supports metadata
  - ISO 19115 elements
  - 3D city model related elements (thematic models, feature count, etc.)

https://github.com/tudelft3d/cityjson/tree/master/schema/v07
Metadata in CityJSON

**Metadata**

The metadata related to the 3D city model may be stored in a JSON object that may have different members, as follows. The members in ISO19115 are used, and a few are added (e.g., presentLoDs, semanticClasses, because they are useful in 3D in a city modelling context). To see all the possible ones, look at the schema file `metadata.json` of a given version.

**"referenceSystem"**

The coordinate reference system (CRS) may be given as a string with an URI to the EPSG code. For instance, for the Dutch national CRS in 3D:

```json
"metadata": {
   "referenceSystem": "urn:ogc:def:crs:EPSG::7415"
}
```

Be aware that the EPSG code should be a three-dimensional CRS, i.e., the elevation/height values should be with respect to a specific datum. It is not possible to give a WKI string with the parameters, or any other way.

**Note:**

Unlike in (City)GML where each object can have a different CRS (e.g., a wall of a building could theoretically have a different from the other walls used to represent the building), in CityJSON all the city objects need to be in the same CRS.

**"bbox" (extent of the dataset)**

While this can be extracted from the dataset itself, it is useful to store it. It may be stored as an array with 6 values: [minx, miny, minz, maxx, maxy, maxz]

```json
"metadata": {
   "bbox": [ 84710.1, 446846.0, -5.3, 84757.1, 446944.0, 40.9 ]
}
```
Energy ADE in CityJSON (coming soon!!)

```json
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "title": "Energy ADE made better, schema v0.7",
    "type": "object",

    "definitions": {

        "ThermalZone": {
            "type": "object",
            "properties": {
                "type": { "enum": ["ThermalZone"] },
                "azimuth": {"type": "number"},
                "inclination": {"type": "number"},
                "area": {
                    "type": "object",
                    "properties": {
                        "units": {"type": "string"},
                        "value": {"type": "number"}
                    }
                },

                "surfaceGeometry": {
                    "type": "array",
                    "items": { "$ref": "#/definitions/MultiSurface" }
                }
            }
        }

        "+EnergyBuilding": {
            "type": "object",
            "properties": {
                "type": { "enum": ["+EnergyBuilding"] },
                "attributes": {
                    "type": "object",
                    "properties": {
                        "creationDate": {"type": "string"},
                        "terminationDate": {"type": "string"},
                        "class": {"type": "string"},
                        "function": {"type": "string"},
                        "usage": {"type": "string"},
                        "measuredHeight": { "type": "number"}
                    }
                }
            }
        }
    }
}
```
citygml4j

citygml4j is an open source Java class library and API for facilitating work with the OGC City Geography Markup Language.
Azul

Azul is a 3D viewer for 3D city models in (City)GML, CityJSON, OBJ, OFF and POLY. It supports loading multiple files, selecting objects by clicking them or selecting them in the sidebar, and browsing their attributes.

What's New in Version 0.8.1
Support for CityJSON 0.5
pip install cjio
Process and manipulate a CityJSON file, and allow different outputs. The different operators can be chained to perform several processing in one step, the CityJSON model goes through the different operators.

To get help on specific command, eg for 'validate':

```
  cjio validate --help
```

Usage examples:

```
cjio example.json validate
cjio example.json removeTextures info
cjio example.json subset --id house12 remove_materials save out.json
```

Options:

- **--version**: Show the version and exit.
- **--off**: Load an OFF file and convert it to one CityJSON GenericCityObject.
- **--ignore_duplicate_keys**: Load a CityJSON file even if some City Objects have the same IDs (technically invalid file)
- **--help**: Show this message and exit.

Commands:

- **compress**: Compress a CityJSON file, ie stores its...
- **decompress**: Decompress a CityJSON file, ie remove the...
- **info**: Output info in simple JSON.
- **merge**: Merge the current CityJSON with others.
- **remove_duplicate_vertices**: Remove duplicate vertices a CityJSON file.
- **remove_materials**: Remove all materials from a CityJSON file.
- **remove_orphan_vertices**: Remove orphan vertices a CityJSON file.
- **remove_textures**: Remove all textures from a CityJSON file.
- **save**: Save the CityJSON to a file.
- **subset**: Create a subset of a CityJSON file.
- **update_bbox**: Update the bbox of a CityJSON file.
- **update_crs**: Update the CRS with a new value.
- **validate**: Validate the CityJSON file: (1) against its...
To sum up...

- Metadata for Energy ADE datasets.
- CityJSON Energy Extension.
- Automatic transformation of CityGML Energy ADE datasets to CityJSON data model.
That’s all folks!

Kavisha – k.kavisha@tudelft.nl