

Attendees

- Alberto Giretti (Università Politecnica delle Marche - UNIVPM)
- Anna Wagner [TU Darmstadt]
- Bjorn Butzin (University of Rostock)
- Gonçal Costa [LaSalle University]
- Kris McGlinn (TCD-ADAPT)
- Mathias Bonduel [KU Leuven]
- Seppo Törmä (VisuaLynk)
- Walter Terkaj (STIIMA-CNR) · ¶
- Wendelin Sprenger [Ed. Züblin AG]
- Aaron Costin (U. Florida)

Date and time

- 11/12/2018
- 14:30 GMT

Agenda

1. "OMG what a FOG - Geometries on the web, A presentation on the Ontology for Managing Geometry and the File format Ontology for Geometry" (Anna & Mathias)
2. Update to progress towards WG

Minutes

- "OMG what a FOG - Geometries on the web, A presentation on the Ontology for Managing Geometry and the File format Ontology for Geometry" (Anna & Mathias)
 - Link to the presentation:
<https://drive.google.com/open?id=12Rvab7-GBY3xQOffpJ9uKprisEPeMdPn>
 - Anna - background product data, multi function facade components. Work based on two projects, SolConPro, SCOPE
 - Mathias - managing heritage data, wide variety of data and stakeholders. Geometry can be point clouds, etc.
 - Geometry currently too central in BIM, should be able to leave out.
 - My work is about static geometry >> parametric geometry.
 - Point clouds, complex meshes, etc.
 - Challenges:

Linking geometries on the web

- Multiple approaches exist, each with benefits and downsides
- Multiple geometry representations (using different approaches) needed addressing the needs of different use cases

But:

- How can we ensure data integrity?
- No unified approach of linking geometry to building objects is defined
- When using non-RDF geometry, how can we include information about the file specific information (e.g. file format, original software, units, up-axis, etc.)?

Implemented supporting ontologies

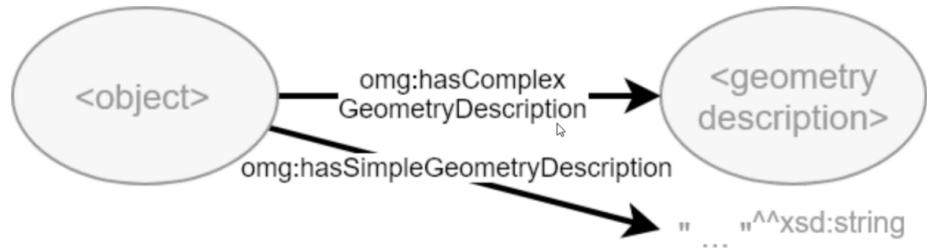
- OMG - Ontology for Managing Geometry
 - Defines relations between building objects and their geometries: uniform way of modeling
 - Allows multiple geometry representations per object
 - Provides means to describe dependencies
 - geometry - geometry
 - geometry - property
 - Versioning and grouping of geometries
 - FOG - File Format Ontology for Geometry
 - Taxonomy of specific relations for connecting geometries per geometry file format / ontology
 - Enhances data exchange by defining geometry formats on relation level
 - Simplifies usage of geometry stored in RDF literals into applications
- Can attach properties to geometry, such as height, to reduce redundancy
- Anna to go into details of the geometries

OMG - basic concept

- small ontology (4 classes, 1 datatype property, 13 object properties)
- re-uses concepts and vocabulary from SEAS, PROV-O, and OPM
- introduces 3 levels for connecting geometry to objects (similar to proposed property modelling - OPM)
 - **level 1**: direct link from object to geometry
 - **level 2**: adding a “Geometry” node between geometry and object for meta-data
 - **level 3**: level 2 + adding a “GeometryState” node for versioning
- provides concepts to describe derivatives of geometries
 - geometry to geometry
 - property to geometry (and geometric properties)
- introduces a grouping mechanism for easier querying (geometry context)



OMG - level 1

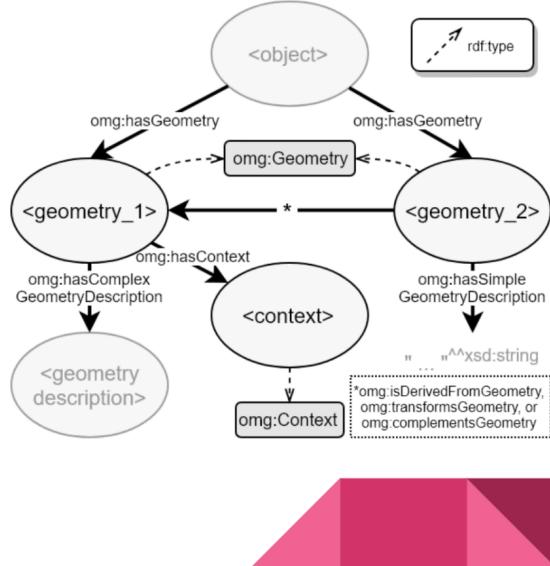


- directly connects geometry to any object
- both simple (non-RDF) and complex (RDF-based) geometries



OMG - level 2

- intermediate 'Geometry' node for meta data
- relations between geometry nodes:
 - derivation: keeping data synched
 - transformation: avoiding unnecessary data redundancies
 - complementation: adding details to already defined geometries
- 'Context' class for grouping
 - simplifies extraction of geometries by context (e.g. planning phase or role)

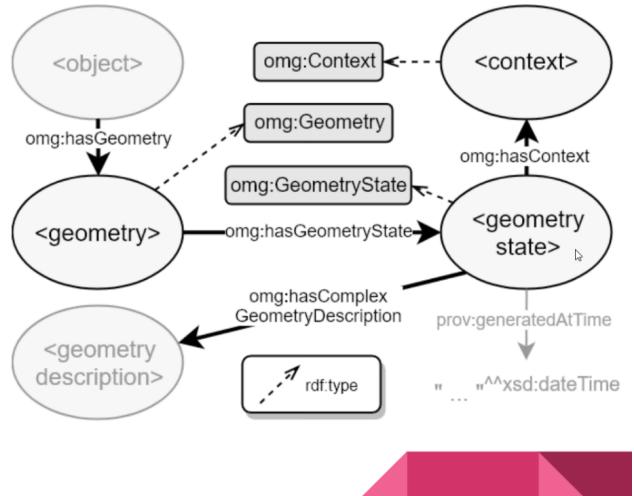


- Geometry context class for extracting based on a context (e.g. planning phase, role, or some specific use case).

OMG - level 3

Overview

- intermediate 'GeometryState' node to add more meta data to the geometry
 - similar to `opm:PropertyState`
 - requires a timestamp of creation
 - `omg:CurrentGeometryState`
 - enables versioning of geometry
 - can also be related to a context

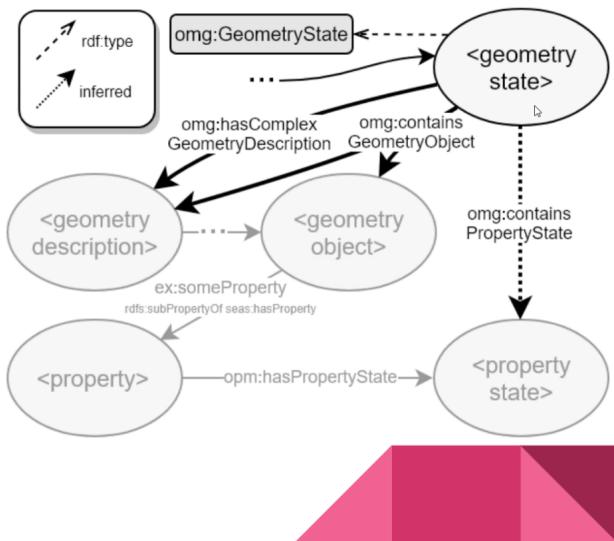


- A geometry may have multiple geometry states, allows for versioning.
- Not feasible to copy graph when single properties are changes. Geometry state should be connected to any geometry object, including geometry description itself.

OMG - level 3

Geometry and property states

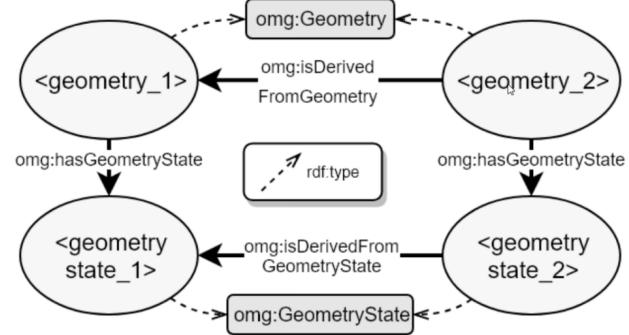
- in case of complex, **RDF-based geometry descriptions**, a geometry state should be connected to any geometry node that might be changed (`omg:containsGeometryObject`)
- chain property infers relation between geometry state and all contained property states



OMG - level 3

Dependent geometry nodes and derived geometry state nodes

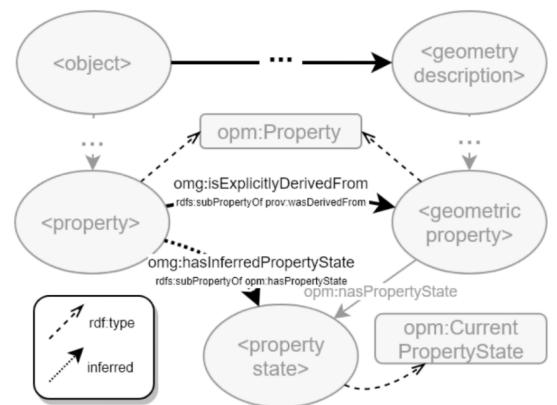
- relation between geometries to define their actual derivation
- can be used to query for outdated geometry descriptions



OMG - Derived properties

Explicitly derived properties

- for **redundant** properties (e.g. height) on semantic and geometric descriptions
- based on OPM property states
- `omg:isExplicitlyDerivedFrom` connects one property to another with the same content
- `omg:hasInferredPropertyState` infers property states (chain)



Levels of geometry

	level 1	level 2	level 3
connecting geometry	yes	yes	yes
multiple geometry representations	no*	yes	yes
defining dependencies between geometries	no	yes	yes
versioning of geometry	no	no	yes
explicitly derived properties	yes	yes	yes
implicitly derived properties	no	yes	yes

o

OMG - Demo

- Namespace: <https://w3id.org/omg#>
 - HTML documentation: <https://w3id.org/omg/>
 - raw ontology: <https://gitlab.iib.tu-darmstadt.de/Wagner/omg.git>

SPARQL visualiser

- <https://madsholten.github.io/sparql-visualizer/?file=https%3F%2Fwww.dropbox.com%2Fs%2Fg1c9oclaxv1l8ud%2Fomg-demo.json>

■

OMG - Summary

- proposes a unified approach for connecting geometry to building data
- enables versioning of geometry
- introduces relations to describe dependencies between geometries, and geometries and properties
- generic relations without information about used geometry formats
- no information about *how* geometries and / or properties are derived
- no definition for linking to geometry description specific meta data
 - up-axis, scale / units, georeferencing (location + orientation), Modeling / exporter / converter software
- not about describing geometry
-
- Mathias Fog

FOG - Basic concepts (1)

- For “simple” (non-RDF) and “complex” (RDF-based) geometry descriptions
- RDF literals and datatypes (non-RDF geometry):
 - RDF literal: ascii and binary (`xsd:string` and `xsd:base64Binary` / `xsd:hexBinary` / ...)
 - External file: absolute or relative file location (`xsd:anyURI`)
- Subproperties of OMG: `omg:hasSimpleGeometryDescription` and `omg:hasComplexGeometryDescription`
- Some file formats define geometry in multiple associated files:
 - OBJ: geometry (.obj) and material (.mtl)
 - glTF: coordination (.gltf), textures (.jpg / .png) and binary geometry (.bin)



FOG - Basic concepts (2)

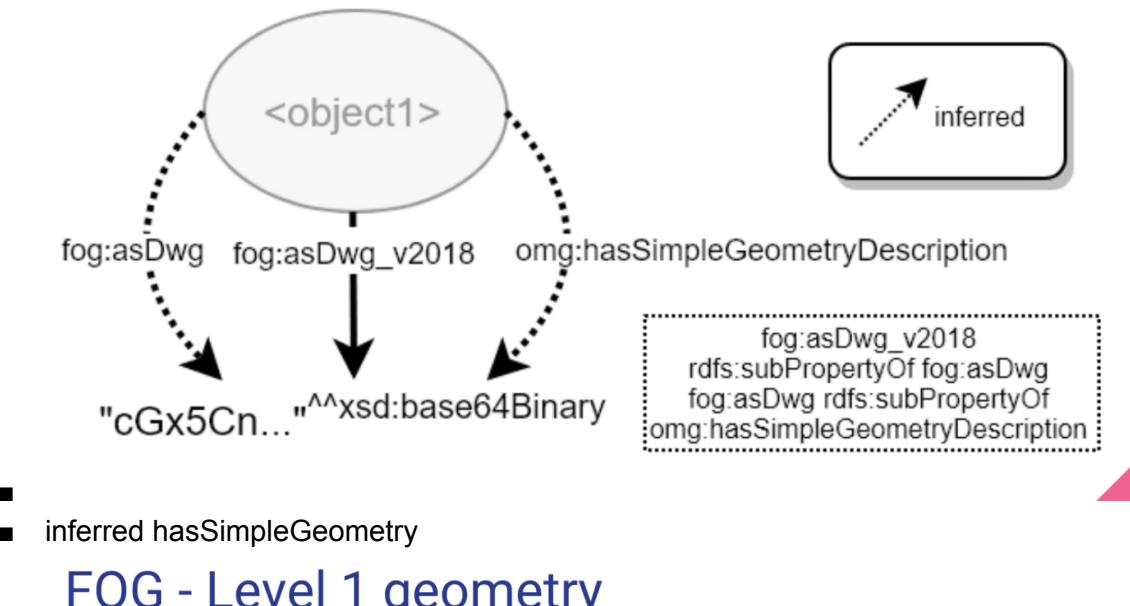
- Taxonomy of properties:
 - file formats & geometry-ontologies
 - format / ontology version
 - file extension
 - (binary LE / BE)
- Metadata per geometry format added in the ontology:
 - Specification(s) / documentation
 - Which file types are associated?
 - If non-RDF: file extension

FOG - Basic concepts (3)

- Taxonomy maintenance should be community effort
 - Uncovered geometry formats
 - New geometry formats
 - Inaccuracies in taxonomy (format versions!)

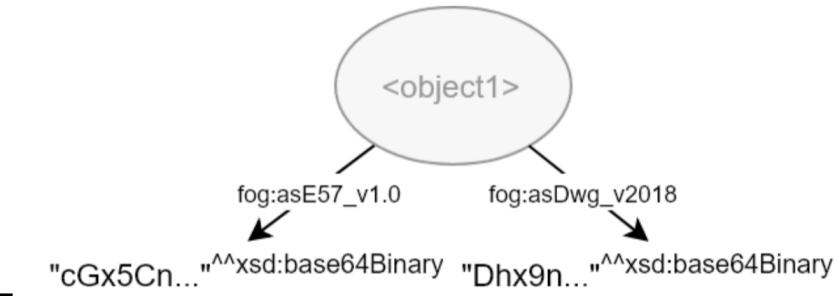
We are on Github: <https://github.com/mathib/fog-ontology>

FOG - Taxonomy

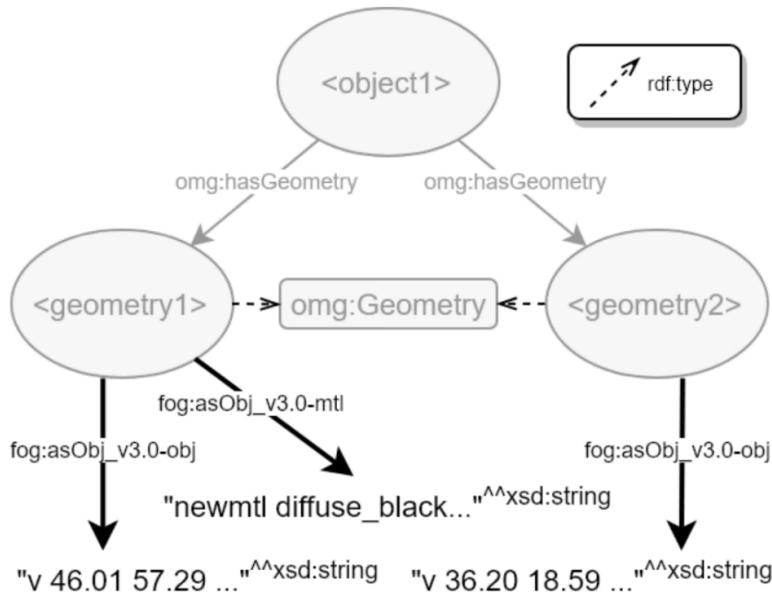


FOG - Level 1 geometry

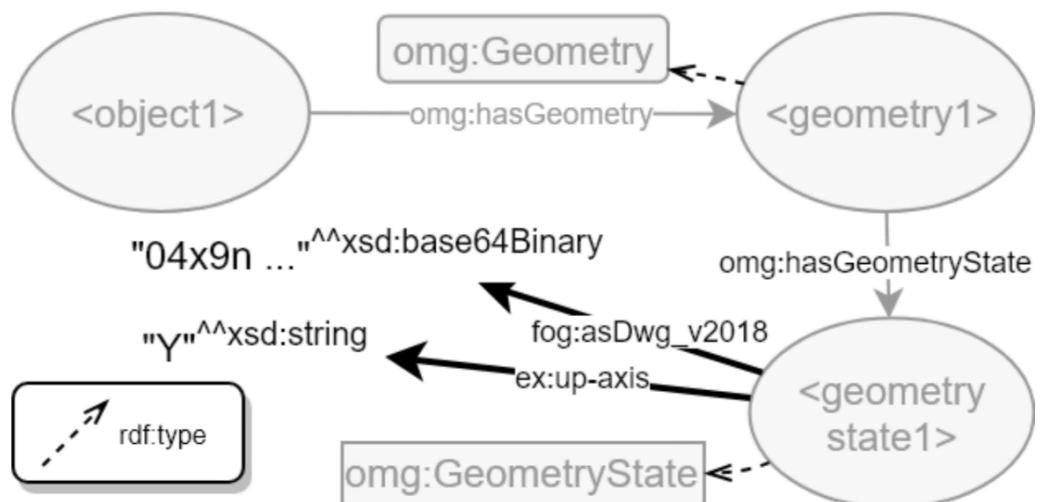
- Cannot be used in case of **multiple** geometry representations of the same format on one building object:
 - Difference between representations?
 - Formats with multiple associated files (e.g. OBJ and glTF)



FOG - Level 2 geometry



FOG - Level 3 geometry



FOG - Demo

- Webviewer for geometry using three.js
- Queries RDF triplestore containing simple and complex geometry
 - Datatype: decode if binary ⇔ fetch if external file
 - Recognises geometry format based on FOG property
 - Visualises (if loader available) in browser
 - All simple geometry can be downloaded to a file
- Future work -
- Alignments between OMG and OPM, BOT
- Enable spatial querying with SPARQL for other geometry besides WKT and GML
- Description of algorithms for derivation of geometries and/or properties on the web
-
- Link to the presentation:
<https://drive.google.com/open?id=12Rvab7-GBY3xQOffpJ9uKprisEPeMdPn>
-
-
- Update to progress towards WG
 - https://drive.google.com/file/d/1U_rDJncAuptZdM2p1rUaVS9ISkaldaZs/view?usp=sharing
 - Suggest to use HTML Editey

Previous minutes

<https://docs.google.com/document/d/1NGN-P1lz9OITbzI6bXHxmPoN52bCzjVPDgPAIMYjdQ/edit?usp=sharing>

Next Call

-