Open Geospatial Consortium

GEOSPARQL 3D WHITE PAPER

TECHNICAL PAPER

DRAFT

CONTENTS

I KEYWORDS

The following are keywords to be used by search engines and document catalogues.

OGC, GeoSPARQL, 3D

PREFACE

To come...

SECURITY CONSIDERATIONS

The following security considerations apply...

IV SUBMITTING ORGANIZATIONS

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

— Organization one

— Organization two

— Organization three

V SUBMITTERS

NAME	AFFILIATION	OGC MEMBER
Nicholas J. Car	KurrawongAI	Yes

1

SCOPE

1 SCOPE

2

CONFORMANCE

2 CONFORMANCE

3

NORMATIVE REFERENCES

3 NORMATIVE REFERENCES

There are no normative references in this document.

4

TERMS AND DEFINITIONS

———

4 TERMS AND DEFINITIONS

No terms and definitions are listed in this document.

5

ABSTRACT

5 ABSTRACT

To come...

KEYWORDS

6 KEYWORDS

To come...

7

CONVENTIONS

7 CONVENTIONS

8

INTRODUCTION

---

8   INTRODUCTION

OGC GeoSPARQL is an open standard that enables storage and exchange of spatial data on the Web, based on the Resource Description Framework (RDF). In 2020 the OGC white paper Benefits of Representing Spatial Data Using Semantic and Graph Technologies was published. It explained the benefits of GeoSPARQL and described the ways in which GeoSPARQL version 1.0 could be improved. In the meantime, GeoSPARQL version 1.1 has been published, having many improvements described in aforementioned white paper, see also this introduction to GeoSPARQL 1.1 [14].

Although not specifically bound, GeoSPARQL 1.1 is mainly geared towards spatial objects having zero to two dimensions. In other words, things that can be displayed on a flat surface. However, for three-dimensional spatial objects both supply and demand are increasing.

Two new versions of GeoSPARQL are in development: 1.2 and 1.3. GeoSPARQL 1.2 will be an adaption of GeoSPARQL 1.1 which enables GeoSPARQL to be published as an ISO standard. Version 1.2 will have no substantive changes with respect to version 1.1. GeoSPARQL 1.3 is expected to have extended capabilities supporting three-dimensional (3D) space. Not only would that make GeoSPARQL more useful for 3D geospatial data, it would also help in industries and knowledge domains that are not mainly focused on geospatial data, like Building Information Modelling (BIM) and Computer Graphics (CG). This paper describes which extensions to GeoSPARQL for 3D are possible, which are most desirable, and how they could be achieved.

This paper consists of three main parts. The first section describes current capabilities of GeoSPARQL: in its current state, GeoSPARQL does allow modelling three-dimensional objects, but with relevant limitations. The second section describes requirements for extended 3D capabilities in GeoSPARQL. This is based on a market consultation, which resulted in a collection of use cases for additional 3D capabilities. By analysing those use cases, as well as current developments within and outside the OGC, insight in the most needed extensions to GeoSPARQL can be obtained. Requirements are then weighed against feasibility; some extensions are easier to achieve than others. The third and final section describes the expected benefits for additional 3D capabilities in GeoSPARQL for different domains and different roles.

This paper should be interesting for the following audiences:

— Current and prospective users of GeoSPARQL;

— Current and prospective implementers of GeoSPARQL;

— Members of related OGC working groups.

A deep understanding of the Semantic Web and graph technologies is not required for readers of this white paper, but some interest in those topics is assumed, as well as familiarity with the aformentioned white paper Benefits of Representing Spatial Data Using Semantic and Graph Technologies. This paper focuses on specifically on 3D capabilities of GeoSPARQL, not its general benefits.

9

RELATED WORK

9 RELATED WORK

This is still just a dump of resources. To be transformed into actual text.

9.1. CityGML

https://www.mdpi.com/2220-9964/12/9/351 IFC-CityGML Data Integration for 3D Property Valuation

[15]

This article describes what became known as Onto CityGML, so far the closest thing to GeoSPARQL3D.

https://www.cambridge.org/core/journals/data-centric-engineering/article/semantic-3d-city-interfacesintelligent-interactions-on-dynamic-geospatial-knowledge-graphs/975E518C41DCDD8BD565012C9E2C8473 Semantic 3D city interfaces—Intelligent interactions on dynamic geospatial knowledge graphs

[10]

Knowledge graphs construction for 3D city data. Related the CityGML and Onto CityGML.

https://link.springer.com/article/10.1007/s41064-020-00095-z CityGML 3.0: New Functions Open Up New Applications

[16]

Looks like the Semantic Web is fully considered in CityGML 3.0. Possibly demanding aligments with GeoSPARQL3D. Question: what can GeoSPARQL add that is not yet in CityGML?

https://lup.lub.lu.se/student-papers/search/publication/9169387 Linked geodata: CityGML represented as a virtual knowledge graph

hansson2024citygml

More knowledge graphs constructed from CityGML.

https://www.sciencedirect.com/science/article/pii/S2666546821000574?via%3Dihub Semantic 3D City Database — An enabler for a dynamic geospatial knowledge graph

[9]

Other details and history of Onto CityGML.

9.2. Medicine & Chemistry

https://www.nature.com/articles/s41597-022-01562-5 Standard metadata for 3D microscopy

[17]

The 3D Microscopy Metadata Standards (3D-MMS), developed by the BRAIN 3D Microscopy Working Group, help ensure that a 3D microscopy dataset is sufficiently described to support its' re-use by scientists who did not generate the data. Adoption of 3D-MMS will aid investigators who want to share data, helping them to evaluate and decide which data can be combined.https://doryworkspace.org/metadata

https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13083 Ontology-Based Representation and Modelling of Synthetic 3D Content: A State-of-the-Art Review

[12]

A range of approaches have been proposed to permit semantic representation and modelling of synthetic 3D content. These approaches differ in the methodologies and technologies used as well as their scope and application domains. This paper provides a review of the current state of the art in representation and modelling of 3D content based on semantic web ontologies, together with a classification, characterization and discussion of the particular approaches.

https://www.inderscienceonline.com/doi/abs/10.1504/IJMSO.2017.087702 A novel ontology for 3D semantics: ontology-based 3D model indexing and content-based video retrieval applied to the medical domain

[13]

This paper presents the most comprehensive formally grounded 3D ontology to date that maps the entire XSD-based vocabulary of the industry standard X3D (ISO/IEC 19775-19777) to OWL 2, complemented by fundamental concepts and roles of the 3D modelling industry not covered by X3D.

9.3. To consider

https://link.springer.com/article/10.1007/s10845-023-02246-6 Ontology of 3D virtual modeling in digital twin: a review, analysis and thinking

[18]

To help novice engineers understand and scheme 3D virtual modeling in digital twin for future research and applications, this paper reviews 106 digital twin 3D modeling cases with their characteristics, including deployment targets, purposes & roles, collaborative models, data flows, the autonomy of 3D modeling, fidelity, twinning rates, enabling technologies, and enabling tools.

STereoLithography (STL)http://www.3dsystems.com/quickparts/learning-center/what-is-stl-file

STL files describe only the surface geometry of a three-dimensional object without any representation of color, texture or other common CAD model attributes. The STL format specifies both ASCII and binary representations.

Open standard for particle-mesh data (openPMD)https://github.com/openPMD/openPMD-standard

The openPMD standard, short for open standard for particle-mesh data files is not a file format per se. It is guidance for meta data and naming schemes. openPMD provides naming and attribute conventions that allow to exchange particle and mesh based data from scientific simulations and experiments. The primary goal is to define a minimal set/kernel of meta information that allows to share and exchange data to achieve portability between various applications and differing algorithms, a unified open-access description for scientific data (publishing and archiving), and a unified description for post-processing, visualization and analysis. If output from programs, devices (such as cameras), simulations or post-processed data-sets contain a minimal set of meta information as provided by openPMD, you can exchange data between those with minimal effort and you use the same tools for visualization.

CARARE Metadata Schemahttps://pro.carare.eu/en/introduction-carare-aggregation-services/carare-metadata-schema/

The CARARE metadata schema is a harvesting schema intended for delivering metadata about an organisation's online collections, heritage assets and their digital resources. The strength of the schema lies with its ability to support the full range of descriptive information about monuments, building, landscape areas and their representations. The CARARE metadata schema builds on existing standards and best practice from a number of different countries in Europe and the rest of the world.

9.4. IFC and BIM

9.4.1. Industry Foundation Classes (IFC) and BIM

BIM is a paradigm in which object-model definitions — with machine-interpretable semantics — are exchanged, rather than relying on CAD drawings that convey only graphical semantics. The predominant open exchange standard is Industry Foundation Classes (IFC).

9.4.1.1. Product model

In IFC, a construction work is decomposed into a set of products. These products can have **multiple representations**. For example, a wall can be described as a solid body as well as a two-dimensional axis. These representations facilitate different views on the same data: an editable line segment or an easily visualized volume. The Object-relational nature of the IFC EXPRESS schema allows intricate relationships such as a representation context that communicates

additional intent for the representation or presentation styles that can be granularly assigned to individual faces.

At the same time, such a product separates the **placement** (an hierarchical transformation) from the actual geometry definition. The consequence of this is that in spite of its object-relational nature, IFC product representations cannot be used for building-level topological relationships between solids, because even if two solids are touching in 3D, the fact the the placement is externalized out of the geometry definition (or the fact that faces are constructed procedurally and do not exist explicitly), means that the two faces cannot be opposite oriented twins. As such, relational geometric constructs such as space boundaries are provided as additional supplementary geometries.

In principle, the IFC schema has been designed in a modular fashion with independent modules for, for example, geometry, materials and meta-data. However in other cases, **semantics and geometry are intertwined** such as tapered extrusions (lofts) where the begin and end profile of a duct carry important semantics.

IFC also allows for **decomposition**, where a whole is aggregated into multiple parts for richer semantics. This allows for example to connect materials and meta-data to the frame and the glazing separately, while still being able to identify the aggregate as a single window. This is not used as frequently, partially due to inability to efficiently instantiate such aggregates as geometry instances.

9.4.1.2. Evolving views on geometry

IFC is heavily influenced by the ISO 10303 (STEP) family of standards, but over time adopted its own geometric paradigms:

— Procedural geometry and boolean operations became less prominent with the adoption of ReferenceView in IFC4. Tessellated geometry definitions were added for more compact exchange.

— **Infrastructure definitions** were added with precise mathematical transition curves and a composition of a horizontal, vertical and cant (inclination) profile.

— IFC5 with an **explicit** (most likely triangulated) geometry schema at the core, with semantic overlays to encode the same procedural semantics as a non-mandatory or use-case specific layer. Heavily inspired by USD with layer-based composition for collaborative exchange.

Especially the handling of **tolerances** means that the standard cannot effectively prescribe a consistent outcome in all cases. Tolerances are needed for BRep model with non-linear underlying geometry and/or fixed precision coordinate values, e.g higher degree nurbs curves are typically intersected with numerical approximation, so a vertex that connects two of such curves needs to have seen as a sphere with the local tolerance as its radius. This tolerance is also applied to boolean operations: an subtraction volume can be slightly inwards of the first operand but is still expected to pierce through the volume and increase surface genus. This contrasts with the desire of using IFC as a legal basis in contracts. NB Tolerances stand in the way of using

existing approaches for SFA geometry predicates such as PostGIS+SFCGAL which is based on arbitrary precision boolean logic as implemented in CGAL without tolerances.

9.4.1.3. Use cases

The most successful use case on BIM data is **coordination and visualization** where multiple aspect models are geometrically overlaid in order to find issues, which are then communicated to the authoring software where they are addressed. This approach works, because it respects that individual disciplines all have their own specialistic software.

Design to design workflows are much harder to realize, although some Model View Definitions have been developed on top of IFC that enable the transfer of design intent in specific and constrained scenarios, such as precast concrete and structural steel.

Long-term preservation of building information is difficult because of the fact that IFC models are difficult to mutate, because they are so explicit and don't contain the vendor-specific design intelligence. Therefore native software cannot always re-import IFC models, but also the native models degrade over time because of the need to migrate to newer editions of the software. Software that can directly operate on IFC to make modifications is still experimental.

BIM-GIS integration is challenging because it requires familiarity with both domains on where to draw the line between euclidean and non-euclidean geometries and acceptable error metrics.

Simulations on IFC building models are often challenging because the 'bag of individual elements' does not provide a good foundation higher order topological representations required for flow-of-energy type of simulations. For e.g thermal simulation a topological view of space boundaries is required. They have been added as secondary set of ternary relationships, but usage of more specific-purpose and simpler schemas sees still more usage in industry. In general, IFC models are created for a specific purpose and wide-spread usage of those models in nieghbouring domaisn remain challenging because modelling for those neighbouring purposes requires alignment on the worldviews and levels of detail that is often beyond the scope in which such models are procured.

9.4.1.4. Implications and questions:

- Euclidean / non-euclidean; is a CRS required?

- Separate representation+placement → enables efficient reinstantiation, but hinders topological relationships because you require the pair of placement+geometry to locate in space

- Geometry as leaf-values or object-relational model : cannot encapsulate geometry into a single literal, but allows for richer semantics

- BRep model (topology + geometry + orientation + location) vs polyhedral model (e.g halfedge) vs explicit loops of point coordinates

- Procedural vs implicit (e.g constraints) vs explicit (polyhedra)

- Tolerances

- Decomposition inside or outside of the 'geometry ontology'

- Are infra geometries (hor + ver alignment + cant, for positioning and sweeps) in scope?

9.5. Implementations

9.5.1. CGAL → SFCGAL → PostGIS

9.5.2. OpenCASCADE

9.5.2.1. OpenCASCADE-inspired BRep ontology

Perzylo, A., Somani, N., Rickert, M., & Knoll, A. (2015, September). An ontology for CAD data and geometric constraints as a link between product models and semantic robot task descriptions. In 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS) (pp. 4197-4203). IEEE.

https://ieeexplore.ieee.org/abstract/document/7353971

[11]

9.5.2.2. Topologic

Jabi, W., & Chatzivasileiadi, A. (2021, January). Topologic: exploring spatial reasoning through geometry, topology, and semantics. In Formal Methods in Architecture: Proceedings of the 5th International Symposium on Formal Methods in Architecture (5FMA), Lisbon 2020 (pp. 277-285). Cham: Springer International Publishing.

https://topologic.app/

[2]

9.5.3. BRep vs mesh/polyhedron

BRep

- Curved surfaces

- Topology: connected components as shells, solids with inner voids, etc.

- Clean APIs due to inheritance: e.g `fn extrude(Topo) → Topo`, for vertex → edge; edge → face; face → solid; solid → solid

- Extra indirections: edge → vertex[] → point

- Depending on implementation can be inefficient, e.g outer wire of face not explicitly marked need to be checked wrt infinite point

- Data integrity and validation a bit harder

Mesh/polyhedron

- Potentially fewer indirections

- Triangle meshes robust and well understood

- Many different data models though, e.g half-edge (only manifold), indexed faceset (no adjacency info), winged/quad/radial edge

10

CURRENT CAPABILITIES

CURRENT CAPABILITIES

10.1. GeoSPARQL

GeoSPARQL is the most common geospatial extension of SPARQL. It was accepted as an OGC standard in 2012 and revised as GeoSPARQL 1.1 in 2024.

According to the standard document, "The OGC GeoSPARQL standard supports representing and querying geospatial data on the Semantic Web. GeoSPARQL defines a vocabulary for representing geospatial data in RDF, and it defines an extension to the SPARQL query language for processing geospatial data".

10.1.1. Requirements addressed

In order to define which capabilities GeoSPARQL needs to adopt for full 3D compatibility, we first take a look at GeoSPARQL 1.1 current capabilities with regards to 3D.

10.1.1.1. Dimensionality

In this white paper, when we discuss three dimensional data, we mean three dimensional geometries; instances of the class geo:Geometry. GeoSPARQL 1.1 defines three different properties for geometry that have to do with dimensionality:

1. geo:dimension, or *topological dimension*: This is the number of perpendicalar directions in which a geometry extends. A point, for example, extends in no direction, so its topological dimension is 0. A line has a length, but no width or height. Its topological dimension is 1. A cube has a topological dimension of 3.

2. geo:coordinateDimension: In the geometric model that GeoSPARQL uses, points are the basic building blocks of geometry. A point geometry can have a different number of coordinates, depending on the space in which it is placed. A point on a flat map, or on the surface of a sphere, needs two coordinates. The *coordinate dimension* is the number of coordinates in the points that define a Geometry.

3. geo:spatialDimension: In order to support linear referencing, geometries in GeoSPARQL can have a measure value. This is a relative position along a line or a curve, and does not extend the Geometry in space. The *spatial dimension*, therefore, can be used to denote the number of coordinates in a point, excluding the measurement value.

The extended capabilities for GeoSPARQL discussed in this white paper concern the cases where the *spatial dimension* (wich can be expressed with property geo:spatialDimension) of geometry has the value 3.

The table below shows the values of the three dimension properties for some Geometries (expressed as Well Known Text (WKT)).

Table 1

GEOMETRY (WKT)	GEO:DIMENSION	GEO:COORDINATEDIMENSION	GEO:SPATIALDIMENSION
point (4 15)	0	2	2
point Z (4 15 3)	0	3	3
linestring (4 15, 13 2)	1	2	2
point M (4 15 60)	0	3	2
point ZM (4 15 3 60)	0	4	3

10.1.1.2. Vocabulary

GeoSPARQL version 1.1 consists of a Core module, a Topology Vocabulary extension, a Geometry extension, a Geometry Topology extension, an RDFS Entailment extension, and a Query Rewrite extension. Each of these modules will be briefly explained below.

10.1.1.3. Core Module

GeoSPARQL version 1.1 includes a CORE module. This defines the basic classes, relationships, and literals. These classes are suitable for both 2D and 3D.

10.1.1.3.1. Topology Vocabulary Extension

This extension provides standard definitions for spatial relationships that can exist between one spatial object and another.

Relations between geometries have been defined using three different sets of rules:

— Simple Features Relations

— Egenhofer Relations

— Region Connection Calculus RCC8

These topological definitions specify topological relations in a 2D-context.

10.1.1.3.2. Geometry Extension

GeoSPARQL 1.1 geometry module defines a class `Geometry` as a subclass of `SpatialObject`.
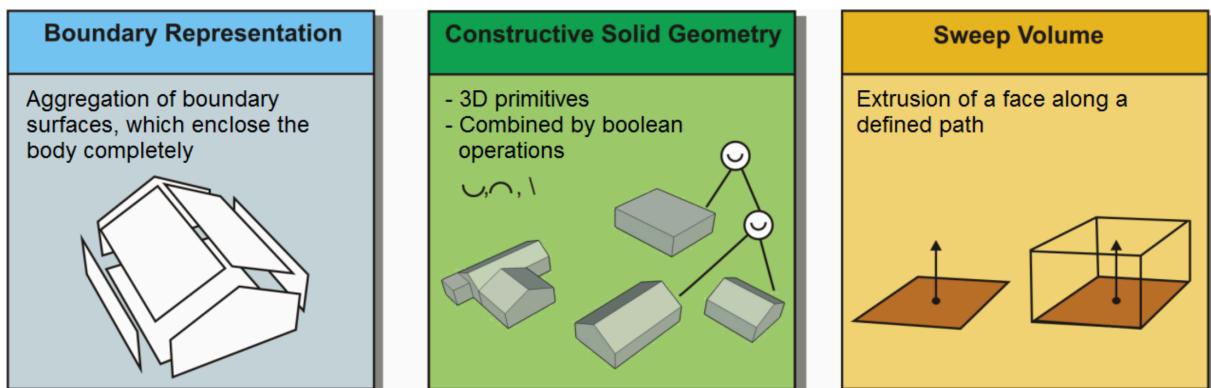
An instance of `Geometry` is not restricted to two dimensions.

A fine-grained classification of `Geometry` can use the <u>Simple Feature Vocabulary</u> which extends the class `Geometry` with further types, such as *Point*, *Polygon* etc.

The Simple Features vocabulary allows for the definition of 3D variants of:

- (Multi)Points

- (Multi)LineStrings

- (Multi)Polygons

- Polyhedralsurface

It does not include commons 3D primitives, such as *Cube* or *Mesh* surfaces which are integral parts of 3D representations. Nor does it include extruded geometry.

Figure 1 — Possible approaches for representing 3D objects in IFC

Concerning metadata of 3D models, GeoSPARQL 1.1 provides properties which can be reused in 3D contexts. In particular, the properties are:
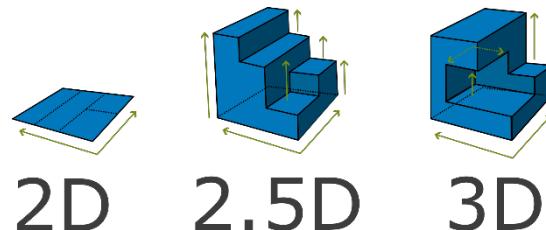
- `geo:hasVolume`

- `geo:hasMetricVolume`

Further 3D-related metadata properties such as projection matrices are not part of the current GeoSPARQL 1.1 standard.

A first requirement for 3D support is the ability to save 3D data in a knowledge graph.

The Geometry extension enables geometries to be defined within the GeoSPARQL framework. Supported geometry formats include RDF literals in WKT, GML, GeoJSON, and KML.

These literals are investigated for the storage of 3D data.

Figure 2 — Difference between 2D, 2.5D and 3D geometry

Table 2

Literal Type	Z-Coordinate Supported	2.5D	3D
WKT Literal	Yes	Yes	Yes
GML Literal	Yes	Yes	Only with extension Schema
KML Literal	Yes	Yes	As import from COLLADA
GeoJSON Literal	Yes	Yes	Yes
DGGS Literal			

GeoSPARQL 1.1 also does not restrict the usage of coordinate reference systems with 3D support. There are currently almost 300 coordinate reference systems in the database epsg.io which can be used to describe 3D data encoded in the GeoSPARQL graph literals listed above.

GeoSPARQL allready supports linking multiple geometries (e.g., 2D and 2.5D) to a single feature. We aim to extend this with 3D geometries.

For example, a tree in the real world may be represented in databases with both a 2D and a 3D geometry.

This should not require duplication of the tree entity — there can be one tree instance with both geometries linked, avoiding redundancy in the data model.

10.1.1.4. Query functions with 3D support

GeoSPARQL 1.1 functions currently do not offer fully-featured 3D support. However, there are functions which may take into account the Z coordinate, if they are available.

Table 3

GeoSPARQL function	Z-Coordinate Supported	2.5D	3D
geof:is3D	Yes	Yes	Yes
geof:minZ	Yes	Yes	Yes
geof:maxZ	Yes	Yes	Yes

These functions check for the presence of Z coordinates or filter out maximum and minimum Z coordinates of the given geometry.

10.1.1.5. Geometry Topology Extension

Another extension is the Geometry Topology extension. This provides query functions that return relationships between different geometries based on the topology vocabulary extension.

10.1.1.6. RDFS Entailment Extension

The RDFS Entailment extension provides rules for reasoning over geometries. Based on specific statements, additional information can be inferred. This kind of logical inference can also be applied to geometry. GeoSPARQL includes logic for reasoning over simple features geometries.

10.1.1.7. Query Rewrite Extension

GeoSPARQL allows queries such as whether "Feature A" is located within "Feature B" using its vocabulary. The Query Rewrite extension specifies a RIF rule that enables query rewriting. However, this extension does not support the rewriting of 3D queries.

REQUIREMENTS FOR GEOSPARQL 3D

REQUIREMENTS FOR GEOSPARQL 3D

11.1. 5.1 Existing implementation of 3D geometry in GeoSPARQL

This section provides an overview of feedback received on the current version of the GeoSPARQL standard (version 1.1) regarding 3D usage. It helps to identify some of the barriers to use, and to outline requirements that have not been addressed that may encourage greater uptake.

11.1.1. Proposed extensions for GeoSPARQL 3D

11.1.1.1. Extension 1: 3D representations

11.1.1.1.1. GitHub Issue URI

https://github.com/opengeospatial/ogc-geosparql/issues/583

11.1.1.1.2. Category

Semantic improvement

11.1.1.1.3. Description

GeoSPARQL should include ways to represent 3D data in a knowledge graph.

3D data should be included in common 3D formats and 3D data should be includable as a text literal and a file representation.

Some common formats which could be considered for inclusion are:

- Polygon File Format (PLY)

- Wavefront OBJ Format (OBJ)

- GLTF Format (GLTF)

- X3D Format

11.1.1.2. Extension 2: Relations of 3D geometries

11.1.1.2.1. GitHub Issue URI

https://github.com/opengeospatial/ogc-geosparql/issues/416

11.1.1.2.2. Category

Semantic improvement

11.1.1.2.3. Description

GeoSPARQL should include ways to represent relations between 3D geometries and relations between 3D geometries and geometries of lower dimensions. The relations should be expressable in property relations and should be queryable using SPARQL extension functions.

11.1.1.3. Extension 3: Appearance of 3D geometries

11.1.1.3.1. GitHub Issue URI

- https://github.com/opengeospatial/ogc-geosparql/issues/584
- https://github.com/opengeospatial/ogc-geosparql/issues/592

11.1.1.3.2. Category

Semantic improvement

11.1.1.3.3. Description

GeoSPARQL should include ways to represent materials and textures of 3D geometries, so that geometries can be styled accordingly.

Materials include:

- Colors of surfaces with light diffusion parameters
- Images as textures, which are associated with surfaces of the 3D object

11.1.1.4. Extension 4: Multi-component 3D geometries

11.1.1.4.1. GitHub Issue URI

https://github.com/opengeospatial/ogc-geosparql/issues/591

11.1.1.4.2. Category

Semantic improvement

11.1.1.4.3. Description

GeoSPARQL should include ways to define multi-component 3D geometries, whereas each component expresses its own semantics. For example, parts of a building could have different semantics according to the function of the building components and would be classified as such in an RDF graph.

11.1.1.5. Extension 5: Positioning of 3D geometries

11.1.1.5.1. GitHub Issue URI

- https://github.com/opengeospatial/ogc-geosparql/issues/587
- https://github.com/opengeospatial/ogc-geosparql/issues/588
- https://github.com/opengeospatial/ogc-geosparql/issues/589
- https://github.com/opengeospatial/ogc-geosparql/issues/591

11.1.1.5.2. Category

Semantic improvement

11.1.1.5.3. Description

GeoSPARQL should include ways to position 3D geometries in a 3D space. Commonly 3D geometries are rotated, translated and scaled using commonly defined operators in computer graphics. Similar operations are needed for the relative positioning of 3D objects in GeoSPARQL, as properties and potentially as functions.

11.1.1.6. Extension 6: Alignments of GeoSPARQL 3D

11.1.1.6.1. GitHub Issue URI

- https://github.com/opengeospatial/ogc-geosparql/issues/590

- https://github.com/opengeospatial/ogc-geosparql/issues/574

- https://github.com/opengeospatial/ogc-geosparql/issues/571

11.1.1.6.2. Category

Semantic improvement

11.1.1.6.3. Description

GeoSPARQL 3D should be aligned to other vocabularies and standard which currently provide 3D support in different knowledge domains. Especially alignments to ifcOWL and the X3D vocabulary would position the GeoSPARQL vocabulary as a link between these different standards.

11.1.1.7. Extension 7: Alignments of Engineering CRS to Geospatial CRS

11.1.1.7.1. GitHub Issue URI

https://github.com/opengeospatial/ogc-geosparql/issues/586

11.1.1.7.2. Category

Semantic improvement

11.1.1.7.3. Description

GeoSPARQL 3D should provide the opportunity to align a local coordinate system in which most 3D geometries are defined with a coordinate reference. While this work might only be partially done within the scope of GeoSPARQL itself, GeoSPARQL should be aligned with the emerging Ontology CRS developments of OGC and provide necessary functions or properties to create the link.

11.1.1.8. Extension 8: Geometry Extrusion

11.1.1.8.1. GitHub Issue URI

— https://github.com/opengeospatial/ogc-geosparql/issues/556

— https://github.com/opengeospatial/ogc-geosparql/issues/547

11.1.1.8.2. Category

Semantic improvement

11.1.1.8.3. Description

GeoSPARQL 3D should provide the opportunity to extrude 2D geometries to 3D geometries and vice versa.

11.1.1.9. Extension 9: Geometry Attributes

11.1.1.9.1. GitHub Issue URI

— https://github.com/opengeospatial/ogc-geosparql/issues/568

— https://github.com/opengeospatial/ogc-geosparql/issues/550

— https://github.com/opengeospatial/ogc-geosparql/issues/549

— https://github.com/opengeospatial/ogc-geosparql/issues/548

— https://github.com/opengeospatial/ogc-geosparql/issues/558

11.1.1.9.2. Category

Semantic improvement

11.1.1.9.3. Description

GeoSPARQL 3D should provide functions and properties that describe essential properties of a 3D Geometry such as its minimum and maximum height, width and depth and its CompactnessRatio.

11.1.1.10. Extension 10: Non-topological Query Functions — 3D Extension

11.1.1.10.1. GitHub Issue URI

— https://github.com/opengeospatial/ogc-geosparql/issues/556

11.1.1.10.2. Category

Semantic improvement

11.1.1.10.3. Description

GeoSPARQL 3D should provide the opportunity to execute non-topological query functions on 2D and 3D geometries commonly used in geospatial databases. Proposed extensions include following functions:

— geometry extrusion to the specified line segment

— geometry extrusion to the specified height

— spatiotemporal geometry extrusion to the specified line segment with specific start and end time

11.2. 5.2 3D Geometry available in IFC

This section describes what kind of geometry is available in IFC, and how that relates to (1) different modelling kernels, and (2) geoSPARQL and geospatial geometry engines.

@Thomas Krijnen, Alex Donkers, Pieter Pauwels == to write here please

11.3. 5.3 Vanilla 3D geometry handling in the Semantic Web

This section describes in what other ways 3D geometry is currently handled in the Semantic Web, for example in BOT ontology, OMG and FOG ontologies, and few more.

11.4. 5.4 Concluding overview of requirements for 3D geometry in the semantic web

A concluding summary with a list of requirements to be taken into account for future development in different places and organisations.

12
BENEFICIARIES AND BENEFITS

12 BENEFICIARIES AND BENEFITS

This section describes the beneficiaries and benefits of representing data, including geospatial data, using semantic and graph technologies. Furthermore, a collection of use cases demonstrate how semantic and graph technologies are used together with spatial data to tackle real world problems.

12.1. Semantic-functional value of 3D-GeoSPARQL

Value for data reasoning, formal description, and semantic interoperability

12.1.1. Meaningful creation and querying of 3D geometry

Whether manually or programmatically defining 3D geometries in BIM, GEO or CG environments, the GeoSPARQL standard could provide or connect to vocabulary to represent 3D structures. A 3D cube, for example, can be represented in multiple ways:

- As a set of bounding points, lines, and faces;

- As a base surface with an extrude function;

- As an 3D geometric primitive (e.g., the concept of a 'cube' as a spatial figure).

Once GeoSPARQL offers the means to define and link 3D geometries in these different ways, it will allow both humans and machines to interpret and interact with the data effectively.

12.1.2. Meaningful creation and querying of 3D topological relationships

Is "3D Geometry A" inside, touching, overlapping, intersecting or above "3D Geometry B"?

Being able to describe and use such topological relationships is crucial for conducting spatial analyses, formulating rules, and deriving knowledge from different heterogeneous datasets.

Beyond describing topological relationships, one should be able to query them:

What is the spatial relationship between 3D Geometry A and 3D Geometry B?

Such queries should return the appropriate topological result (e.g., "intersects"), which is essential for use cases like clash detection in design validation

12.1.3. Derivation of geometric and topological knowledge

From explicitly modeled geometric and topological data, one should be able to infer implicit knowledge.

For instance, if 3D Geometry A is 3D inside 3D Geometry B, then we can infer that:

A does not "touch" B, and A does not "overlap" with B,

even if these facts are not explicitly stated.

For entailment an application that is compliant with the geo-spatial 3D specification would be able to answer queries like: "Does this tree have a 3D geometry?"

If the tree is defined using a MultiSolid geometry, then the answer should be "yes," even if this is not explicitly declared as such.

12.2. Value for user experience, visualization, and application use of 3D-GeoSPARQL

12.2.1. Visualization

There is a growing need to visualize how 3D data in BIM and GIS relate to one another. Stakeholders want to see BIM and/or 3D Geo data of a newly planned structure visualized within the 3D Geo and/or BIM context of the existing digital city. A shared vocabulary that can present both domains in an integrated way supports this goal.

12.2.2. Calculation and analysis

Semantic modelled 3D-model enables powerful computation and analysis. Analytical results can be displayed in dashboards operating within the GeoBIM domain. This creates a bridge between asset management (typically GIS-oriented) and project management (typically BIM-oriented), allowing for cross-domain collaboration and decision-making.

12.2.3. Spatial querying

Spatial querying is a key function within a federated GeoBIM Network. Questions such as "What material is located in this area?" or "Where is space available for new cables and pipelines?" are examples of 3D spatial queries that can be answered when GeoSPARQL 3D is functioning effectively across systems and semantics.

12.2.4. Machine control

A shared semantic GeoBIM Network also supports machine control. The GIS domain typically describes the existing situation, while the BIM domain describes the intended or future situation. This combination can be used to automatically instruct and guide machines in the built environment.

12.2.5. Modeling, simulation and planning

A network of datasets with 3D-data should support the modeling, simulation and planning of future changes to the built environment. Users — whether human or machine — can use the network and GeoSPARQL 3D to propose and model modifications in either BIM or GIS formats. A well-functioning 3D semantic GeoBIM Network enables this kind of forward-looking spatial planning and design.

12.3. Added value of the 3D-GeoSPARQL for Organisations

12.3.1. Brings together different domains (GIS, BIM, CG)

GeoSPARQL 3D can help in bringing together different domains that work with 3D information.

12.3.2. Enables extension to vocabularies from other domains

The vocabulary can have the possiblity to extend to vocabulaires with 3D geometry or topology from other domains, for example the Building Ontology Topology (BOT) or RELOC Ontology. Multi-domain models with georeferenced 3D and 2D city models could be also linked by CityRDF ontology based on CityGML 3.0 standard. GeoSPARQL 3D would be important enabler for data retrieval, update and analysis for such models. In addition to ontologies, it can also establish relationships to 3D file formats from other domains, such as glTF.

12.3.3. Facilitates better interoperability between knowledge domains with spatial components (geography, aerospace, architecture, product design, (bio)chemistry, IoT, ...)

A unified geometric language can help by migrating ideas and methods across fields, accelerating innovation. For example a parametic design algoritm, rule, or llm making use of geometry and topology originated in aerospace, can be re-used in the architecture domain.

13
ANNEX N: N

13 ANNEX N: N

14

ANNEX O: HISTORY

14 ANNEX O: HISTORY

BIBLIOGRAPHY

BIBLIOGRAPHY

[1]   Abhayaratna J, van den Brink L, Car N, Atkinson R, Homburg T, Knibbe F, McGlinn K, Wagner A, Bonduel M, Rasmussen MH, Thiery F: Abhayaratna2020wp, *OGC Benefits of Representing Spatial Data Using Semantic and Graph Technologies*. Open Geospatial Consortium (2020). http://www.opengis.net/doc/wp/using-semantic-graph.

[2]   Jabi W, Chatzivasileiadi A: Topologic: Exploring Spatial Reasoning Through Geometry, Topology, and Semantics. In: p. 277. Springer International Publishing, Cham. (2021).

[3]   Filip Biljecki HT: QUALITY OF BIM–GIS CONVERSION. (2019).

[4]   Anne Göbels JB: Relative Location Ontology: An ontological model for representing directional topological relationships between spatial entities in oriented space. (2024).

[5]   Donkers S, Ledoux H, Zhao J, Stoter J: Automatic conversion of IFC datasets to geometrically and semantically correct CityGML LOD3 buildings. *Transactions in GIS* vol. 20, p. 547 (2016).

[6]   Hansson F: Linked geodata: CityGML represented as a virtual knowledge graph. *Student thesis series INES* (2024).

[7]   *World Wide Web Consortium: RDF 1.1 Concepts and Abstract Syntax, W3C Recommendation*, https://www.w3.org/TR/rdf11-concepts/ (25 February 2014).

[8]   *World Wide Web Consortium: RDF 1.1 Turtle Terse RDF Triple Language, W3C Recommendation*, https://www.w3.org/TR/turtle (25 February 2014).

[9]   Chadzynski A, Krdzavac N, Farazi F, Lim MQ, Li S, Grisiute A, Herthogs P, von Richthofen A, Cairns S, Kraft M: Semantic 3D City Database — An enabler for a dynamic geospatial knowledge graph. *Energy and AI* vol. 6, p. 100106 (2021).

[10]   Chadzynski A, Li S, Grišiūtė A, Chua J, Hofmeister M, Yan J, Tai HY, Lloyd E, Tsai YK, Agarwal M, Akroyd J, Herthogs P, Kraft M: Semantic 3D city interfaces—Intelligent interactions on dynamic geospatial knowledge graphs. *Data-Centric Engineering* vol. 4 (2023).

[11]   Perzylo A, Somani N, Rickert M, Knoll A: An ontology for CAD data and geometric constraints as a link between product models and semantic robot task descriptions. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 4197–4203. IEEE. (2015).

[12]   Flotyński J, Walczak K: Ontology-Based Representation and Modelling of Synthetic 3D Content: A State-of-the-Art Review. *Computer Graphics Forum* vol. 36 no. 8, pp. 329–353 (2017).

[13]   Sikos LF: A novel ontology for 3D semantics: ontology-based 3D model indexing and content-based video retrieval applied to the medical domain. *International Journal of Metadata, Semantics and Ontologies* vol. 12 no. 1, p. 59 (2017).

[14]   Car NJ, Homburg T: GeoSPARQL 1.1: Motivations, Details and Applications of the Decadal Update to the Most Important Geospatial LOD Standard. *ISPRS International Journal of Geo-Information* vol. 11 no. 2, p. 117 (2022).

[15]   El Yamani S, Hajji R, Billen R: IFC-CityGML Data Integration for 3D Property Valuation. *ISPRS International Journal of Geo-Information* vol. 12 no. 9, p. 351 (2023).

[16]   Kutzner T, Chaturvedi K, Kolbe TH: CityGML 3.0: New Functions Open Up New Applications. *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science* vol. 88 no. 1, pp. 43–61 (2020).

[17]   Ropelewski AJ, Rizzo MA, Swedlow JR, Huisken J, Osten P, Khanjani N, Weiss K, Bakalov V, Engle M, Gridley L, Krzyzanowski M, Madden T, Maiese D, Mandal M, Waterfield J, Williams D, Hamilton CM, Huggins W: Standard metadata for 3D microscopy. *Scientific Data* vol. 9 no. 1 (2022).

[18]   Wang Y, Wang X, Liu A, Zhang J, Zhang J: Ontology of 3D virtual modeling in digital twin: a review, analysis and thinking. *Journal of Intelligent Manufacturing* vol. 36 no. 1, pp. 95–145 (2023).