Towards a domain specific graph query language for the geoscience - implementing a GeoGQL -

Markus Wilhelm Jahn[1 2 3]

[1]KIT, Karlsruhe Institute of Technologie, 76131 Karlsruhe, Germany - markus.jahn@kit.edu
[2]UFZ, Helmholtz Centre for Environmental Research, 04318 Leipzig, Germany - markus.jahn@ufz.de
[3]OGC, Open Geospatial Consortium, 22201 Arlington, USA - mjahn@ogc.org

Keywords: graph query language, geometrically induced topology; polytope complexes; topological algorithms; watertight volumetric model; spatio-temporal geo-data;

Abstract

Modern geo-data management plays a crucial role in designing digital twins in distributed system environments, enabling seamless integration, analysis, and visualization of spatial information. With the rise of graph databases and linked data, geospatial relationships can be efficiently modeled and queried using technologies such as Gremlin, a graph traversal language. On the other hand, Simple Features (see ISO19107) and the Dimensionally Extended 9-Intersection Model (DE-9IM) as two traditional examples provide standardized frameworks for spatial representation and topological reasoning, ensuring interoperability across systems. The fusion of geospatial standards with schema-free geo-data management advances the support of real-time decision-making and scalable geospatial applications, making modern geo-data management a cornerstone of intelligent, interconnected digital environments. Giving meaning to standards by ontologies is one major rapprochement to establish semantic interoperability. This paper provides one step towards this goal by using an abstract graph schema to represent the intra- and inter-relations of simplicial- and polytope-complexes and applying the traditional geoinformatics interpretation of topology, the philosophy of the Dimensionally Extended 9-Intersection Model (DE-9IM) to give meaning. This approach can be seen as a step towards the implementation of a Domain-Specific-Language (DSL) for property graphs that represent complex interrelated vector data within a linked data world.

1. Introduction

Distributed systems are composed of multiple interconnected computing nodes that work together to achieve a common goal. Unlike centralized systems, they distribute data, processing, and resources across different locations, enhancing scalability, fault tolerance, and performance. In geo-data management, distributed systems enable efficient storage, retrieval, and processing of spatial data across multiple servers or cloud environments. This is particularly important for digital twins, where real-time geo-data integration and analysis are critical. By leveraging distributed architectures, geo-applications can handle large-scale, complex datasets while ensuring consistency, reliability, and high availability in dynamic environments.

FAIR modeling follows the FAIR principles - Findability, Accessibility, Interoperability, and Reusability - to ensure that data and models are well structured, machine-readable, and reusable across different domains and systems. In the context of geo-data management and digital twins, FAIR modeling promotes standardized data structures, semantic linking, and open formats, enabling seamless integration and sharing across distributed systems. By incorporating graph databases, linked data, and formalized spatial representations such as the Dimensionally Extended 9-Intersection Model (DE-9IM), FAIR modeling enhances geo-data discoverability and usability, supporting scalable and interoperable applications in scientific research or generally the private or public sector.

Property graphs are a flexible data model used to represent complex, interconnected data. They consist of nodes (entities), edges (relationships), and properties (key-value pairs) that can be attached to both nodes and edges. This structure allows for efficient querying and traversal of relationships, making property graphs well-suited for modeling geo-data. For example, Gremlin, a graph traversal language, enables efficient navigation and analysis of property graphs as a implementation of a Graph-Query-Language (GQL) (GQL ISO, n.d.). A standard for a GQL was first described in ISO/IEC 39075, released in April 2024 by ISO/IEC. A Domain-Specific-Language (DSL) is a specialized language tailored to a specific application domain. A DSL provides expressive and optimized querying capabilities for domain-specific data structures.

In geo-applications, a DSL can facilitate topological queries, spatial reasoning, and semantic integration, bridging traditional concepts like the Dimensionally Extended 9-Intersection Model (DE-9IM) with modern, schema-free geo-data management. The development of GeoSPARQL (GeoSPARQL, n.d.) by the OGC (Open Geospatial Consortium, n.d.) and its application to topological relationships in geo-data is a significant area of research in the Semantic Web. The paper "GeoSPARQL: Enabling a Geospatial Semantic Web" (Battle and Kolas, 2011) provides vocabulary and query mechanisms for expressing topological relationships (e.g., intersects, within) in SPARQL queries. The OGC White Paper on "Benefits of Representing Spatial Data Using Semantic and Graph Technologies" (OGC Benefits of Representing Spatial Data Using Semantic and Graph Technologies, n.d.) highlights the advantages of using semantic technologies like GeoSPARQL for representing and querying geo-data, focusing on how such approaches improve data integration, interoperability, and reasoning across diverse geospatial datasets. Additionally, the paper "GeoSPARQL 1.1: Motivations, Details and Applications of the Decadal Update to the Most Important Geospatial LOD Standard" (Car and Homburg, 2022) discusses the key updates and new features in Geo-

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-4-W6-2025-97-2025 | © Author(s) 2025. CC BY 4.0 License.

SPARQL 1.1, further refining its ability to handle more complex spatial data and topological queries. GeoSPARQL is merly designed for knowledge graphs and not property graphs. Table 1 illustrates some differences.

The data model and geometric algorithms of *DB4GeOGraphS* (Jahn et al., 2017, Breunig et al., 2016) were applied. Many ideas of *DB4GeOGraphS*'s data model are introduced in (Jahn and Bradley, 2021, Jahn and Bradley, 2022a, Jahn et al., 2022). In addition to the data model and geometric algorithms, new access methods based on space filling curves (Bradley and Jahn, 2020) and a topological access method (Jahn and Bradley, 2022b) have been investigated. Most of which are parts of the dissertation (Jahn, 2022).

This paper focuses on geometrically induced topology managed by the concept of property graphs and overlays the traditional GIS concept of the Dimensionally Extended 9-Intersection Model (DE-9IM) to give semantic meaning in order to come one step closer towards a DSL for geo-data, a Geo-Graph-Query-Language (GeoGQL). Therefor, vocabularies or grammar are not provided at this stage. The focus is on the underlying pattern matching queries of a GQL with a well defined property graph schema. Comparisons and benchmarks on different topics will be provided in following papers.

2. Data Model

A query language depends on a data model. As in object relational database management systems (ORDBMS) such as *PostgreSQL* with *PostGIS* (PostgreSQL, n.d., Le et al., 2013, H.H., 2014, Le et al., 2014, Weihed, 2015, Gabriel et al., 2015) where a column type may be a complex object-oriented geo-data type, one can think of object property graph databases where a property can be an object of some complex object-oriented geo-data type, too. This section briefly reintroduces the underlying data model developed to manage spatial and spatio-temporal data, which follows the object property graph approach implemented in *DB4GeOGraphS* (Jahn, 2022).

Parts of the geo-graph schema were inspired by the *OGC*s feature model, ISO 19107 and 19109, where a so-called "feature" represents a real-world object. Nodes within a geo-graph can be seen as features if they provide special properties, e.g. a spatial part, a temporal part, and a thematic part as know from the *OGC*s (Open Geospatial Consortium) feature model. The term "feature" instead of "node" to not confuse the geo-community will be used in the following, since a GeoGQL is designed for those users in the first place and not for the informatics-community neither the mathematics-community.

The following definition of a feature summarizes its properties:

Definition 2.1 A feature is a tuple $f = (s, t, a)$ where:

1. s is the spatial part;
2. t is the temporal part;
3. a is the thematic part.

The spatial part of a feature has a dimension (e.g. *Sample*, *Curve*, *Surface* and *Volume*). Features may also move and morph over time. This is done by extruding simplices along the temporal axis to polytopes, following the *Polthier and Rumph* model (Polthier and Rumpf, 1994). Therefore, as an example, a moving curve is actually a surface in spatio-temporal space. Eventually, I do not call, for example, a moving and morphing curve a surface in order to not confuse the geo-community, here too, since a GeoGQL should be designed for the geo-community. However, the polytopes can be aggregated to sequences to form spatio-temporal polytope complexes which represent a moving and morphing spatial simplex. Those sequences can be aggregated to represent moving and morphing simplicial complexes. Figure 1 illustrates the different complexes for each spatial dimension. If the model does not involve moving and morphing, then the simplex are grouped to pure spatial simplicial complexes. A set of complexes is called a net, which is the topological sum of the complexes. This model is designed to be intuitive to ease the handling of complex vector-based features.

The temporal part of a feature consists of temporal entities comparable to the spatial model, but in 1-dimensional temporal space instead of the 3-dimensional spatial space to go along with the spatial or thematic model. That means, in case of a "$3D$+$1D$"-model (features which move and morph over time) for each feature each polytope - the temporal extrusion of a simplex - of the spatial part is linked to a temporal interval of the temporal part. What we get is a moving simplex where the first simplex is linked to the first timestep and the second simplex is linked second timestep of the temporal interval. This representation is a boundary representation of a moving and morphing simplex (Jahn and Bradley, 2021). In case of a "$3D$+$0D$"-model (features which do NOT move and morph over time), which manages different topologically independent snapshots of some spatial model, for each feature each simplex of the spatial part is related to a timestamp of the temporal part. Those relations between the spatial and the temporal parts enable spatio-temporal online analytic processing (ST-OLAP). On the other hand, relations between the thematic and the temporal part can be thought of the same way if the thematic model defines a set of elements where each element can be related to a temporal timestamp or interval. Those relations, on the other hand, enable temporal online analytic processing (TOLAP). Putting both together, we could speak of spatio-temporal temporal online analytic processing (ST-TOLAP).

Since the management of features ought to be done by a property graph, the possible relations between and within the features are of great importance. The relation types where inspired by the object oriented programming paradigm and the mathematical interpretation of topology for simplicial complexes. I defined three different bidirectional types. The relation types are (a) the bidirectional abstraction relation which divides in the unidirectional relations *generalization-of* and *specialization-of*, (b) the bidirectional aggregation relation which divides in the unidirectional relations *part-of* and *composite-of* and (c) the bidirectional incidence relation which divides in the unidirectional relations *boundary-of* (links a boundary-feature into the direction of an interior-feature e.g., an edge with its bounded faces) and *interior-of* (links an interior-feature into the direction of a boundary-feature e.g., a face with its bounding edges). Those relation types define the geometrical induced topologies, the intra- and interrelations of complex features through the geo-graph. Those relations are called geo-relations in the following.

Figure 1 (bottom) illustrates the boundaries of the spatio-temporal polytope complexes of figure 1 (top). The complexes

Feature	GeoSPARQL	GeoGQL
Primary Use	RDF-based *knowledge graphs*	*Property graphs*, potentially RDF-compatible
Standardization	OGC and W3C-aligned	OGC (planned)
Base Language	SPARQL	GQL family (e.g. Cypher, ...)
Data Model	RDF triples (subject–predicate–object)	Property graph model (nodes, relationships, properties)
Semantics	Strong semantics with ontology support	Procedural, less semantic emphasis
Spatial Operations	Rich set of topological and metric functions (e.g. `within`, `distance`)	Similar operations planned, adapted for property graphs
Reasoning Support	Yes – supports entailment regimes	Not a focus – primarily querying
Status	Established (since 2012, updated in 2021)	Emerging
Typical Use Cases	Semantic web GIS, linked spatial data	Geospatial analytics in property graph databases

Table 1. Comparison of GeoSPARQL and GeoGQL

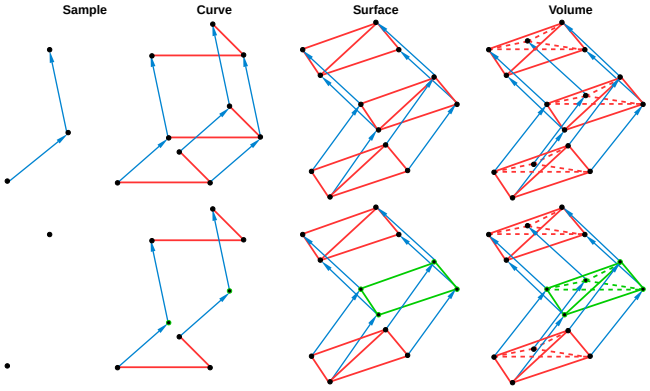

Figure 1. Top: Moving and morphing spatial simplicial complexes as polytope complexes (blue arrows indicate a movement, red lines indicate a line segment); Bottom: Boundary polytope complexes from polytope complexes

of both figures are related by the incidence relations, *boundary-of* and *interior-of*. Having boundary representations (*BREPs*) in mind, the geo-community is more used to the idea that the boundary of a feature is the feature itself or, in a weaker sense, a *BREP* is enough to define a feature. The last is not true if the dimension of the feature is smaller than the spatial or spatio-temporal space it is living in. A constraint needs to be added to define the interior points. For example, a 2-dimensional surface defined by its 1-dimensional boundary as a *BREP* (e.g. walls within CityGML LOD2) lies on a 2-dimensional hyperplane within a d-dimensional space with $d > 2$, in other words it needs to be planar. The definition of hyperplanes to describe the interior points is a common way to define lower-dimensional features. However, I focus on incidence relations and the definitions of interior, boundary and exterior to explicitly distinguish this matter within the structure of the provided geo-graph. This concept implies, that the interior of some boundary is open, like an open interval on some coordinate axis. They are open-sets known from the mathematical theory of topology and their closure is defined by their boundary. The boundary is closed, since it contains all of its limit points.

Definition 2.2 A geo-relation is a tuple $r = (f, f', p)$ where:

1. f is the source feature;
2. f' is the target feature;
3. $p \in \{$*boundary-of*, *interior-of*, *part-of*, *composite-of*, *specialization-of*, *generalisation-of*$\}$ is a label describing the relation type.

Since I concentrate only on features and their geo-relations, the geo-graph is defined as follows:

Definition 2.3 A geo-graph is a tuple $g = (F, R)$ where:

1. F is a finite set of features $f_0, ..., f_m$ with $m > 0$;
2. R is a finite set of geo-relations $r_0, ..., r_n$ with $n > 0$.

For the exploration of the full potential of property graphs, the general definition of property graphs needs to be taken into account. The relations and nodes may carry multiple labels, multiple sets of labels or any properties, including geo-properties, usually managed by key-value pairs where a key identifies a property. As an example, a car x (feature) was produced 1982 in (semantic relation with temporal property) factory y (feature), or the woman "Bettisia Gozzadini" (feature) graduated in the year 1237 at University of Bologna (semantic relation with a geo-property) with a law degree (node). We could also integrate other relations, maybe from a meta-data concept or anything else and combine those to a more specialized Domain-Specific-Language (DSL). But I concentrate on nodes being features, see definition 2.1, in order to model complex geo-objects with their geometrically induced intra- and inter-relations, see definition 2.2, and call the graph a geo-graph, see definition 2.3, in order to add, step by step, the necessary elements for a common GeoGQL. This approach should also provide the potential to glue to other domains in the context of GQLs as a general modular approach.

3. Meaning by the *Dimensionally Extended* 9*-Intersection Model* (DE-9IM)

As mentioned above, traditional features are now presented by two different features, (a) the interior-feature and (b) its boundary-feature. This, as a matter of fact, influences the algorithms and results when calculating intersections or differences of features or any common spatial or spatio-temporal predicate operations such as touches, covers and so on. Figure 2 illustrates the DE-9IM together with the intersections of the exteriors of each boundary. Distinguishing explicitly between interior- points and the boundary-points as two different spatial or spatio-temporal objects by definition has the advantage to split the DE-9IM. This leads to more robust algorithms, since the model itself is smaller for each feature, leaving out redundant elements, redundant in the sense of a feature being a combination of the interior and its boundary. Figure 3 illustrates the intersection model of two features with outsourced boundary-intersections due to the fact that I explicitly distinguish between the boundary-feature and the interior-feature.

The incidence information of the DE-9IM is kept implicitly within the structure of the geo-graph, which turns the evaluation of relations of traditional features into a union of relation evaluations of their interior-features and boundary-features where

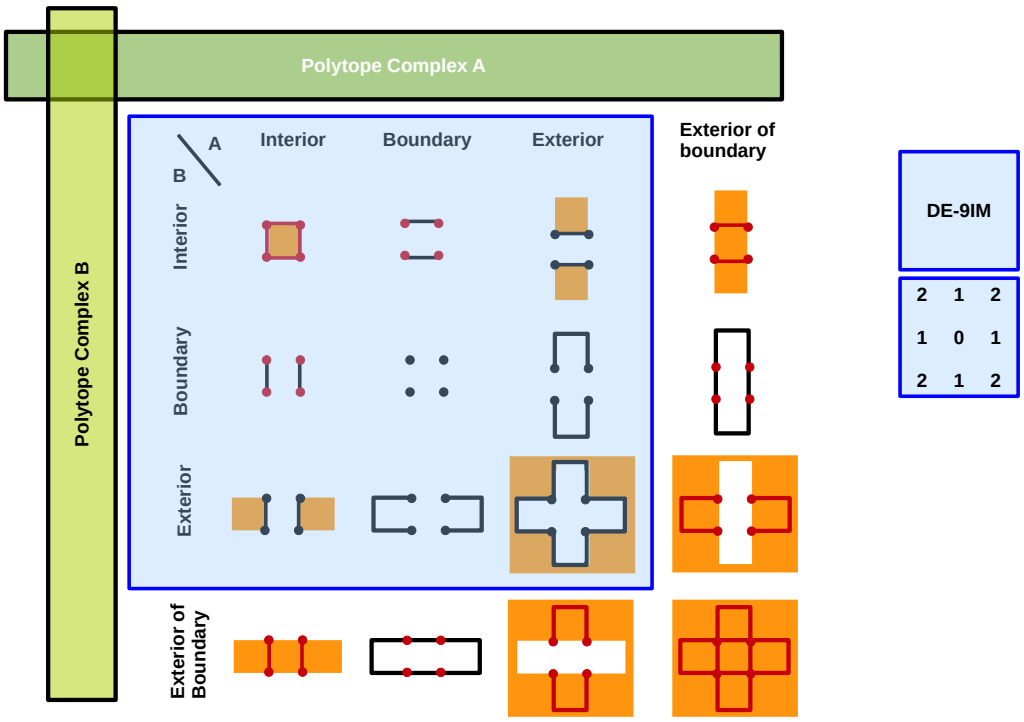

Figure 2. Dimensionally Extended 9-Intersection Model (DE-9IM) of two overlapping rectangles (green and dark-green) together with the intersections of the exteriors of each boundary. The surface (2d) intersections are orange, the curve (1D) and sample (0D) intersections are black. Red indicates geometries which are not part of the intersections.

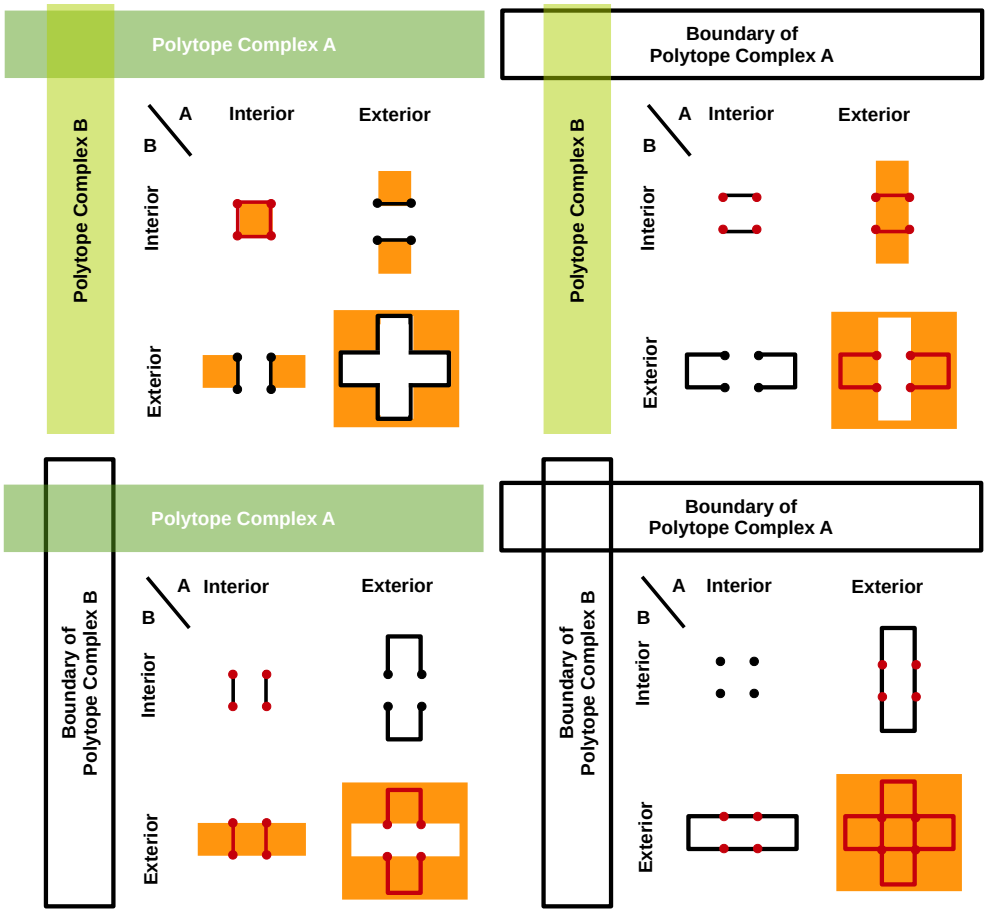

Figure 3. Same as 2 but sorted differently. The model is smaller for each feature to feature intersection since the boundary-intersections are outsourced to their own intersection matrix.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-4-W6-2025-97-2025 | © Author(s) 2025. CC BY 4.0 License.

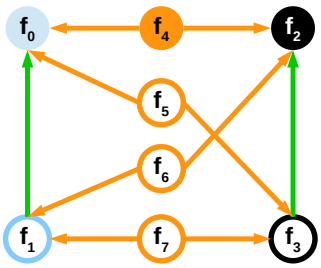

Figure 4. geo-graph with green *boundary-of*-relations, orange *part-of*-relations and identifiers for the example query 3.1

the evaluation of relations turn to pattern matching queries on the geo-graph where the union of evaluations can be expressed by single vocabularies (e.g. intersects) of a DSL, the GeoGQL.

Notice that Figures 2 and 3 show the same patterns, except that for illustration purposes the matrices are sorted differently. The four captions in the first matrix become pairs of two identical sets of captions in the second matrix. The reason is, that the interior complexes A and B and their boundary complexes are treated as individual topological spaces in the second case (Figure 3), while in the first case the boundary was considered as one of the intersection parameters (Figure 2). This change of viewpoint allows one to only consider the intersection parameters interior and exterior because the boundary is now considered as a topological space of its own. However, the exterior of a boundary complex is not reflected in the original DE-9IM, which is the reason why it is contained in Figure 2 and 3 together with the DE-9IM.

If we want to test the geo-predicate *EQUALITY* for two features within an spatial or spatio-temporal object property graph (geo-graph), we can use the geometric equality operation of the spatial part of the involved features. The computational geometry algorithm can be a time consuming process. But since the intersection-features have been calculated to create a geometrically induced topological consistent geo-graph beforehand, we can filter possible candidates by checking the existence of certain intersection-features, reducing the number of time consuming processes. Figure 4 illustrates the possible intersection-features between two features f_0 and f_2 with their boundary-features f_1 and f_3. If f_4 and f_7 exists, equality is possible. If f_5 or f_6 exists, equality is not possible, since one boundary lies inside of the other feature. Looking at the geo-predicate *DISJOINT* it is easy to understand, that there should not be any intersection-features. In case of the geo-predicate *INTERSECTS* there should be at least on intersection-feature $f_{4...7}$. The geo-predicate *TOUCHES* allows only one intersection-feature between the boundary-features e.g. f_7. The geo-predicates *CROSSES*, *OVERLAPS*, *WITHIN* and *CONTAINS* depend on an intersection-feature f_4 between the interior-features. They are different in how the boundary-features intersect. In case of *WITHIN* and *CONTAINS* it is necessary that the boundary-feature of the feature which lies within the other feature intersects the interior-feature of the other feature only, so f_7 does not exist and either f_5 or f_6 exists. The cases *CROSSES* and *OVERLAPS* are not distinguishable from another by the geo-graph structure only. Further geometric analysis need to be done as in case of equality tests.

The formulation of a common FROM - MATCH - WHERE - RETURN query using the identifiers of figure 4 is as follows:

Query 3.1 List all *DISJOINT* features
FROM g
MATCH (f_1:feature) - [:*boundary-of*] → (f_0:feature),
MATCH (f_3:feature) - [:*boundary-of*] → (f_2:feature),
MATCH (f_4:feature) - [:*part-of*] → (f_0:feature),
MATCH (f_4:feature) - [:*part-of*] → (f_2:feature),
MATCH (f_5:feature) - [:*part-of*] → (f_0:feature),
MATCH (f_5:feature) - [:*part-of*] → (f_3:feature),
MATCH (f_6:feature) - [:*part-of*] → (f_1:feature),
MATCH (f_6:feature) - [:*part-of*] → (f_2:feature),
MATCH (f_7:feature) - [:*part-of*] → (f_1:feature),
MATCH (f_7:feature) - [:*part-of*] → (f_3:feature)
WHERE f_0 <> f_2
AND f_4 IS NULL
AND f_5 IS NULL
AND f_6 IS NULL
AND f_7 IS NULL
RETURN f_0,f_1

This query returns a list of all disjoint features. We can remove parts of the WHERE statement in order to find intersection-features as written above. If we remove "f_5 IS NULL" and "f_7 IS NULL" as an example, we would list all features where f_0 could be within f_2 or f_2 could contain f_0.

4. Implementation

I am developing a *TinkerPop* PlugIn to implement the domain specific GQL for the geoscience, a GeoGQL. As a byproduct for testing purposes, I use the *DB4GeOGraphS* framework (Jahn, 2022) to read CityGML data and elevation data to create a topologically consistent space by the algorithms provided in (Jahn and Bradley, 2021) and to create the topologically consistent object property graph of *DB4GeOGraphS*, the geo-graph. When the geo-graph is created I export certain spatial parts of the features to a *VTK* (Visualization Toolkit, n.d.) data format in order to visualize spatial or spatio-temporal results with *ParaView* (ParaView, n.d.) and import the geo-graph into the gremlin server of *TinkerPop*. *TinkerPop* does not support properties being complex objects. Efforts need to be taken to register new object types which may not be integrable in any *TinkerPop*-enabled backend. I follow a different approach for testing purposes.

The features get imported without their spatial-, temporal- and thematic-part. But the level of detail to which extend the simplicial complexes are going to be represented by the property graph of *TinkerPop* can be adjusted. It is possible to create a property graph which includes the whole intra- and inter-relations of each feature together with all location information of all points where each point becomes a interconnected feature within the graph itself. In that case, every information of any features spatial part is represented by the features sub-graph and there is no need for an object property graph and the primitive types of the node properties are enough. However, if we do not pursue such an fine grained level of detail when translating the geo-graph to a primitive property graph we loose the location information and/or other topological information of the simplicial complexes which narrows the usages.

A geometrically induced topologically consistent space implies that redundancies are removed and intersections are properly generated and linked using the provided graph schema. *DB4GeOGraphS* offers three equality tests for spatial objects

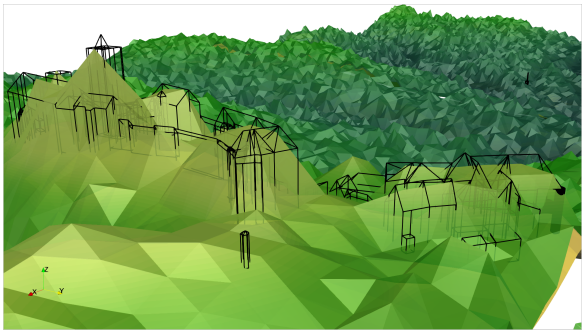

Figure 5. Digital Surface Model (DSM), Digital Terrain Model (DTM) and the raw CityGML wirframe.

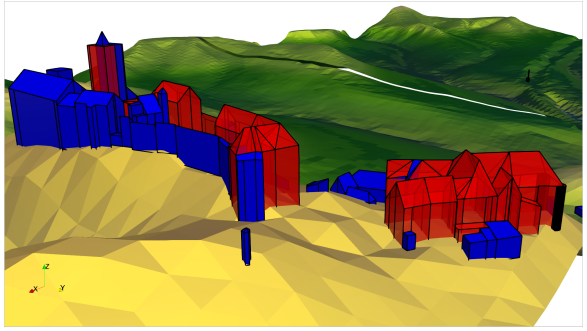

Figure 6. DTM with triangulated CityGML (red) and tetrahedralized CityGML (blue).

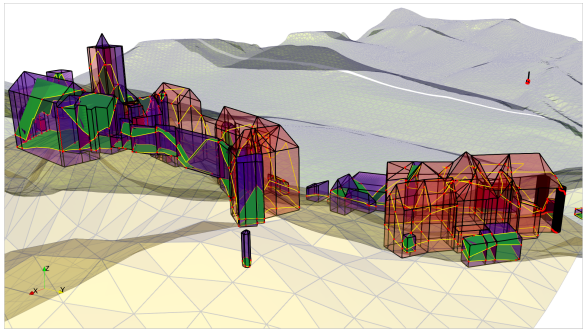

Figure 7. Intersections of CityGMLs wirframe- , surface- and solid-model with DSM and DTM, points (red), curves (yellow) and surfaces (green).

which are referential equality, topological equality and geometric equality (Jahn et al., 2022) where referential equality compares the memory address, topological equality compares the simplices, and geometric equality compares the shapes. Geometric equality is independent of the used tetrahedralisation algorithm, or triangulations algorithm or in terms of curves, if multiple segments can describe one segment, then the two curves are equal spatial objects. I use the last equality type, since this is the most general one. The precision model of *DB4GeOGraphS*, on the other hand, is based on epsilon-environments (Jahn, 2022). It includes test for planarity, angle equalities, skewness etc. The building method of the geo-graph takes care about those redundancies and links the intersecting parts to their compositions. Spatial and spatio-temporal access methods, and the mentioned equality tests are used to check if a spatial or spatio-temporal part already exist and reuses features instead of creating new ones. As a result, for example, the CityGML wireframe model of LOD2 gets reduced to a redundant free spatial model while creating the complete topology graph for each BREP (e.g. walls, roofs, grounds, ...).

This influences the quality of possible intersection-features. For example, the test for equality does not need to be performed since redundant features are not possible. In case of the geo-predicates *WITHIN* and *CONTAINS*, we just need to check if the boundary-feature and interior-feature are completely part of the other interior-feature and not part of its boundary-feature which means, that there is a direct *part-of* relation of the covered feature to the covering feature without any *part-of* relations to the boundary of the covering feature. The cases *CROSSES* and *OVERLAPS* are not distinguishable from another by the geo-graph structure only. Further geometric analysis need to be done.

As a test scenario, I used open geo-data provided by the Thuringian state of Germany (Geoportal Thuringia, n.d.), which includes the Wartburg — a castle where Martin Luther translated the Bible into German in the 16th century. Guided by the principles of Renaissance humanism and driven to combat the corruption of the wealthiest and powerful, Luther aimed to instigate meaningful change, ultimately giving rise to the Protestant Church. Figures 5, 6 and 7 show the Wartburg to get an impression of the test dataset and the generated spatial data. I use the Graph Notebook from Amazon Web Services (graph-notebook, n.d.) to visualize graph queries and compare them with the exports to *VTK* (Visualization Toolkit, n.d.) for visualization in *ParaView* (ParaView, n.d.). These can then be applied to the *TinkerPop* Plugin for implementing the DSL, GeoGQL.

Query 4.1 shows some meta data of the example dataset. The nearby city Eisenach is partly contained besides the Wartburg, also. Each node carries some primitive properties if applicable (e.g. hyper-volume, dimension and complexity) calculated by *DB4GeOGraphS* during the pre-processing step. But, the dimension of a geo-object is also retrievable by filtering one of its simplices and count the edges in the path of one of its points to that simplex using only *boundary-of*-relations. So if the simplex is a triangle we could query the path: (point) - [*boundary-of*] → (segment) - [*boundary-of*] → (triangle). This makes two *boundary-of* steps so the dimension is two. This can be done only for geo-objects which simplices are equal in dimension. The complexity of a geo-object can also be identified by analyzing the geo-objects internal sub-graph. Same is true when querying the hyper-volume of an geo-object if the hyper-volumes of each simplex is present or calculateable. But a detailed discussion of how to calculate spatial properties by analyzing the graph structure would overburden the paper. All together, it is to say that most of the interesting spatial properties are query-able only if the full graph is available (with all simplices and points) ideally in the most topologically consistent representation. The last of Query 4.1 shows how to count intersections only by using the number of geo-objects some geo-object belongs to, in this case equal to two.

Query 4.1 Metadata using TinkerPop Gremlin

```
g.V().count(); 715649
g.E().count(); 1631938
g.V().hasLabel('GEOMETRY').count(); 566425
g.V().has('NODE', 'NAME','Building').count(); 953
g.V().has('COMPLEXITY', 'COMPONENT')
.has('DIMENSION', 'SOLID3D').count(); 1046
g.V().hasLabel('GEOMETRY')
.where(__.out("PART_OF").count().is(2)).count(); 57913
```

Figures 8, 9 and 10 show parts of the graph filtered by the us-

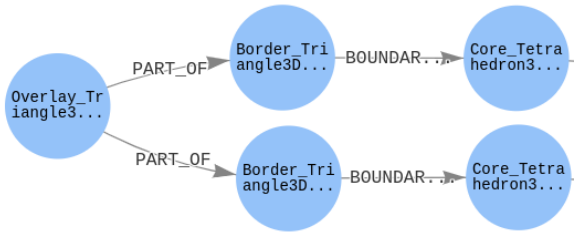

Figure 8. Graph of two tetrahedralized building models. The boundaries intersect.

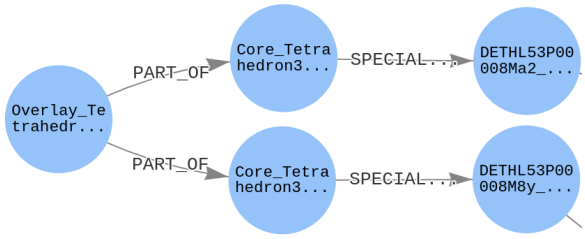

Figure 9. Graph of two tetrehedralized building models. The interiors intersect.

age of the Graph Notebook from Amazon Web Services (graph-notebook, n.d.). Figure 9 shows the graph of two intersecting building interiors. Typically the interiors of two buildings should not intersect. That is a typical problem due to computational geometry error or faulty datasets. As mentioned before, all nodes carry the hyper-volume-property which can be used to find out how large a shared wall between buildings is or even how large the volume of a city can be ((Jahn and Bradley, 2021)).

5. Discussion

The provided data model includes the definition of a spatio-temporal property graph, the geo-graph, which node properties are complex geometric objects, in terms of object-oriented programming. This has upsides and downsides, comparable to the differences of relational database management systems (RDBMS) versus object relational database management systems (ORDBMS). As for *PostgreSQL* with *PostGIS* extension, the benefit of using spatio-temporal data types as column types lies in the benefit of developing processes together with data- and index- structures by the object oriented paradigm and integrate them into the data management system which supports object oriented data types. I followed the same approach when developing *DB4GeOGraphS*. The implementation of the simple feature access specification of the OGC (Simple Feature Access, n.d.) in an object oriented manner to extend the property graph paradigm to an object property graph paradigm let

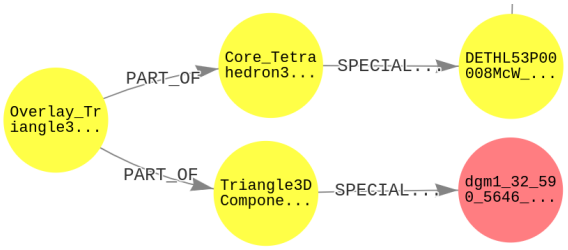

Figure 10. Graph of a tetrahedralized building model with a digital terrain model (DTM). The interiors intersect.

to the development of a *GeoGQL* specification. While I could use the internal geo-operations, data- and index-sturctures of *DB4GeOGraphS* to manipulate the existing geo-graph, problems have arisen on how to integrate that into property graph frameworks like *TinkerPop* which are focused on primitives as properties.

As written in section 4 the geo-graph gets translated into a *TinkerPop* graph which does not support complex geo-datatypes being properties of nodes by default. We loose the spatial information in that case and it is not possible to keep the *TinkerPop* graph geometrically induced topological consistent by itself nor using spatial or spatio-temporal access methods to find specific features. But, as mentioned before, on cost of memory, it is possible to translate the geo-graph to a *TinkerPop* graph where each simplex is topologically integrated using the graph schema of *DB4GeOGraphS*. Since spatial and spatio-temporal access methods most likely are trees they can be integrated into primitive property graphs also. It is in question if this approach is efficient and if there is a need for complex geo-objects being properties of nodes as in an object property graph paradigm.

With *DB4GeOGraphS*s data model as an implementation of an object property graph paradigm together with its graph schema it is possible to analyze geo-predicates by matching certain graph patterns using FROM - MATCH - WHERE - RETURN queries. The complexity of those queries reduces to pattern matching algorithms without the need of recalculating geometric intersections since the topological information needed to query geo-predicates is presented by the property graph explicitly. The same can be done with spatio-temporal data types enabled within ORDBMS (e.g. *PostGIS* with *PostgreSQL*) by creating all geometrically induced topologies managed by different tables and by reformulating the geo-predicates of the simple feature access specification using a well-defined entity-relationship diagram based on the provided graph schema. It is in question which geo-data may be processed more efficiently through which backend type. I will address spatio-temporal indexing, comparison of object-property graphs vs. primitive property graphs and benchmarking in additional papers.

6. Conclusion

Through my previous work, a geo-graph with a certain schema was derived which manages features by adding the needed feature properties to the node definitions of the geo-graph. Furthermore, a pattern matching query was derived to filter nodes in order to test certain geo-predicates reducing the number of computational geometry processes. Graph manipulation, chains of graph transformations or graph projections are in the focuses when developing a standard GQL. This is still work in progress but the focus on the *Dimensionally Extended 9-Intersection Model* (DE-9IM) gave some insights on how the model works and adds the common geo-scientific meaning to it.

The interesting part is how the geo-operations such as calculating intersections or differences of features can be expressed by the graph query language and finally be calculated by the backend. To adapt the simple feature access specification of the OGC (Simple Feature Access, n.d.) new vocabularies where introduced which label the geo-operations to extend SQL or SPARQL. In my case, the standard *GeoGQL* specification is not yet developed. Research needs to be taken on how geo-operations and access-methods should be included in a standard *GeoGQL* specification and, furthermore, how they can be

integrated into the backend. Is there a need for object property graphs in order to efficiently handle semi-structured spatio-temporal data or are property graphs enough which properties are based on primitive types only? My future research aims to answer this general questions by testing the integration of the spatio-temporal data types, spatio-temporal access methods and the used computational geometry algorithms by the implementation of a *DB4GeOGraphS TinkerPop* PlugIn which may extend a *TinkerPop* graph by spatio-temporal property types, geo-operations and spatial or spatio-temporal access methods if more efficient in some situations.

7. Acknowledgments

I would like to thank the Free State of Thuringia ((Geoportal Thuringia, n.d.)), Germany, for making the spatial data publicly available. I gratefully acknowledge the support of the Federal Republic of Germany for funding this research through the project "Distributed Simulation of Processes in Buildings and City Models" (Project No. 469999674).

My sincere gratitude goes to Patrick Erik Bradley, Mulhim Al Doori, and Martin Breunig for their invaluable support and the many constructive discussions that significantly contributed to the progress of my study.

References

Battle, R., Kolas, D., 2011. Geosparql: enabling a geospatial semantic web. *Semantic Web Journal*, 3(4), 355–370.

Bradley, P., Jahn, M., 2020. On the Behaviour of p-Adic Scaled Space Filling Curve Indices for High-Dimensional Data. *The Computer Journal*.

Breunig, M., Kuper, P., Butwilowski, E., Thomsen, A., Jahn, M., Dittrich, A., Al-Doori, M., Golovko, D., Menninghaus, M., 2016. The Story of DB4GeO - A Service-Based Geo-Database Architecture to Support Multi-Dimensional Data Analysis and Visualization. *ISPRS Journal of Photogrammetry and Remote Sensing*, 117, 187–205.

Car, N. J., Homburg, T., 2022. GeoSPARQL 1.1: Motivations, Details and Applications of the Decadal Update to the Most Important Geospatial LOD Standard. *ISPRS International Journal of Geo-Information*, 11(2). https://www.mdpi.com/2220-9964/11/2/117.

Gabriel, P., Gietzel, J., Le, H. H., Schaeben, H., 2015. Gst: A network based datastore for geoscience data and geomodels and its implementationpromin. . . s contribution towards interoperability.

Geoportal Thuringia, n.d. https://geoportal.thueringen.de/gdi-th/download-offene-geodaten visited 2025.

GeoSPARQL, n.d. https://www.ogc.org/publications/standard/geosparql/ visited 2025.

GQL ISO, n.d. https://www.iso.org/standard/76120.html visited 2025.

graph-notebook, n.d. https://github.com/aws/graph-notebook visited 2025.

H.H., L., 2014. Spatio-temporal Information System for the Geosciences: Concepts, Data models, Software, and Applications. PhD thesis, Technische Universität Bergakademie Freiberg.

Jahn, M., 2022. Distributed & Parallel Data Management to Support Geo-Scientific Simulation Implementations. PhD thesis, Karlsruhe Institute of Technology.

Jahn, M., Bradley, P., 2021. Computing watertight volumetric models from boundary representations to ensure consistent topological operations. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, VIII-4/W2-2021, 21–28. https://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/VIII-4-W2-2021/21/2021/.

Jahn, M., Bradley, P., 2022a. A Robustness Study for the Extraction of Watertight Volumetric Models from Boundary Representation Data. *ISPRS International Journal of Geo-Information*, 11(4). https://www.mdpi.com/2220-9964/11/4/224.

Jahn, M., Bradley, P., 2022b. Topological Access Methods for Spatial and Spatiotemporal Data. *ISPRS International Journal of Geo-Information*, 11(10). https://www.mdpi.com/2220-9964/11/10/533.

Jahn, M., Bradley, P., Al Doori, M., Breunig, M., 2017. Topologically consistent models for efficient big geo-spatio-temporal data distribution. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-4/W5, 65–72.

Jahn, M., Kuper, P., Breunig, M., 2022. EFFICIENT SPATIO-TEMPORAL MODELLING TO ENABLE TOPOLOGICAL ANALYSIS. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, X-4/W2-2022, 137–144. https://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/X-4-W2-2022/137/2022/.

Le, H. H., Gabriel, P., Gietzel, J., Schaeben, H., 2013. An object-relational spatio-temporal geoscience data model. *Computers Geosciences*, 57, 104-115.

Le, H. H., Schaeben, H., Jasper, H., Görz, I., 2014. Database versioning and its implementation in geoscience information systems. *Computers Geosciences*, 70, 44-54.

OGC Benefits of Representing Spatial Data Using Semantic and Graph Technologies, n.d. https://docs.ogc.org/wp/19-078r1/19-078r1.html visited 2025.

Open Geospatial Consortium, n.d. https://www.ogc.org/ visited 2025.

ParaView, n.d. https://www.paraview.org/ visited 2025.

Polthier, K., Rumpf, M., 1994. A concept for time-dependent processes. Goebel et al. (ed.), *Visualization in Scientific Computing*, Springer, Vienna, 137–153.

PostgreSQL, n.d. https://www.postgresql.org/ visited 2025.

Simple Feature Access, n.d. https://www.ogc.org/publications/standard/sfa/ visited 2025.

Visualization Toolkit, n.d. https://vtk.org/ visited 2025.

Weihed, P., 2015. *3D, 4D and Predictive Modelling of Major Mineral Belts in Europe*. Ed. Springer International Publishing.