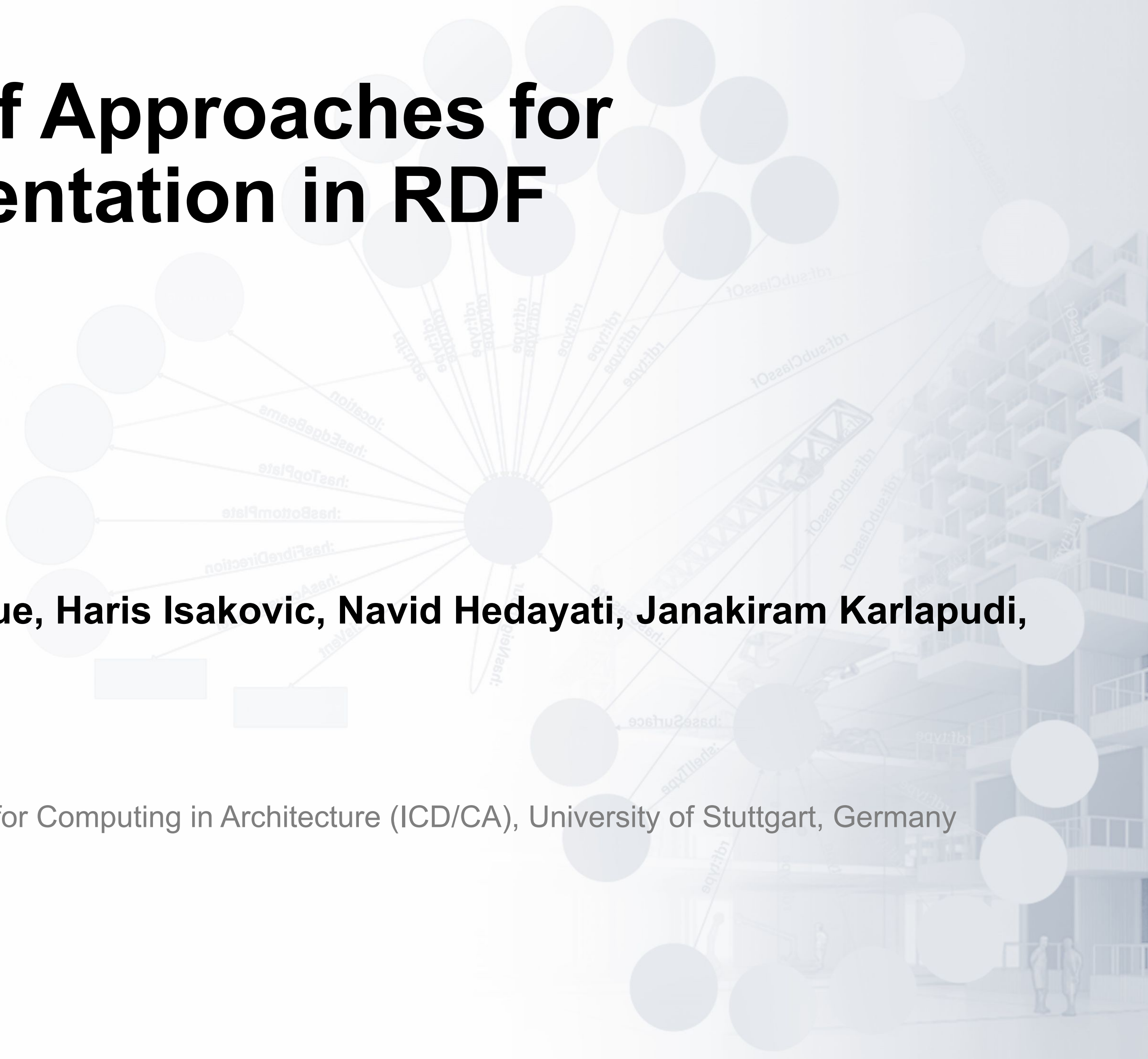
Comparative Analysis of Approaches for Geometric Data Representation in RDF

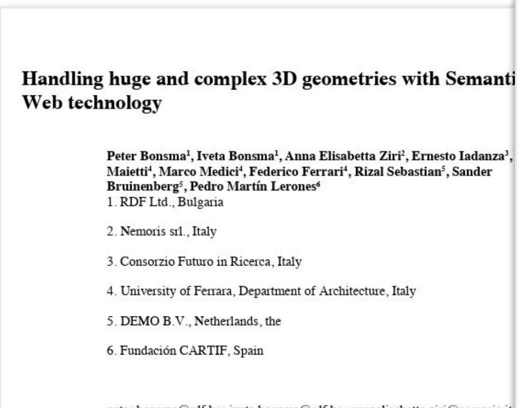


Diellza Elshani, Ali Nakhaee, Anthony A. Arrascue, Haris Isakovic, Navid Hedayati, Janakiram Karlapudi, Thomas Wortmann

Institute for Computational Design and Construction, Department for Computing in Architecture (ICD/CA), University of Stuttgart, Germany
GROPYUS Technologies GmbH, Berlin, Germany

3D Geometry and Semantic Web

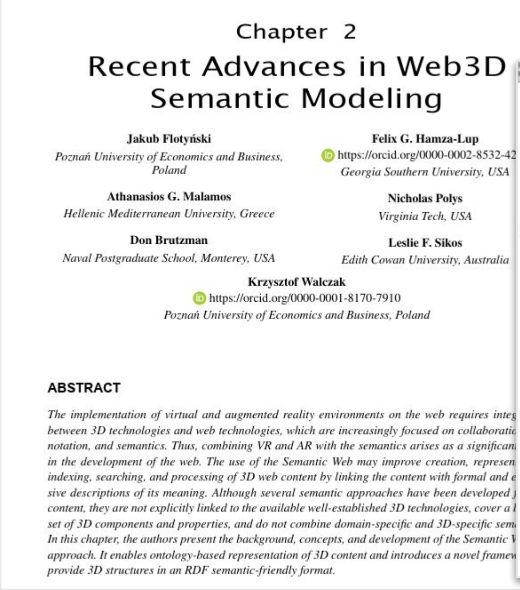
3D Geometry and Semantic Web



Handling huge and complex 3D geometries with Semanti
Web technology

Peter Bonsma[1], Iveta Bonsma[1], Anna Elisabetta Ziri[2], Ernesto Iadanza[3],
Maietti[4], Marco Medici[4], Federico Ferrari[4], Rizal Sebastian[5], Sander
Bruinenberg[5], Pedro Martín Lerones[6]
1. RDF Ltd., Bulgaria
2. Nemoris srl., Italy
3. Consorzio Futuro in Ricerca, Italy
4. University of Ferrara, Department of Architecture, Italy
5. DEMO B.V., Netherlands, the
6. Fundación CARTIF, Spain

peter.bonsma@rdf.bg; iveta.bonsma@rdf.bg; annaelisabetta.ziri@nemoris.it



Chapter 2

Recent Advances in Web3D
Semantic Modeling

Jakub Flotyński
Poznań University of Economics and Business, Poland

Felix G. Hamza-Lup
https://orcid.org/0000-0002-8532-42
Georgia Southern University, USA

Athanasios G. Malamos
Hellenic Mediterranean University, Greece

Nicholas Polys
Virginia Tech, USA

Don Brutzman
Naval Postgraduate School, Monterey, USA

Leslie F. Sikos
Edith Cowan University, Australia

Krzysztof Walczak
https://orcid.org/0000-0001-8170-7910
Poznań University of Economics and Business, Poland

ABSTRACT

The implementation of virtual and augmented reality environments on the web requires integ
between 3D technologies and web technologies, which are increasingly focused on collaboratio
notation, and semantics. Thus, combining VR and AR with the semantics arises as a significan
in the development of the web. The use of the Semantic Web may improve creation, represen
indexing, searching, and processing of 3D web content by linking the content with formal and e
sive descriptions of its meaning. Although several semantic approaches have been developed f
content, they are not explicitly linked to the available well-established 3D technologies, cover a l
set of 3D components and properties, and do not combine domain-specific and 3D-specific sem
In this chapter, the authors present the background, concepts, and development of the Semantic V
approach. It enables ontology-based representation of 3D content and introduces a novel framew
provide 3D structures in an RDF semantic-friendly format.

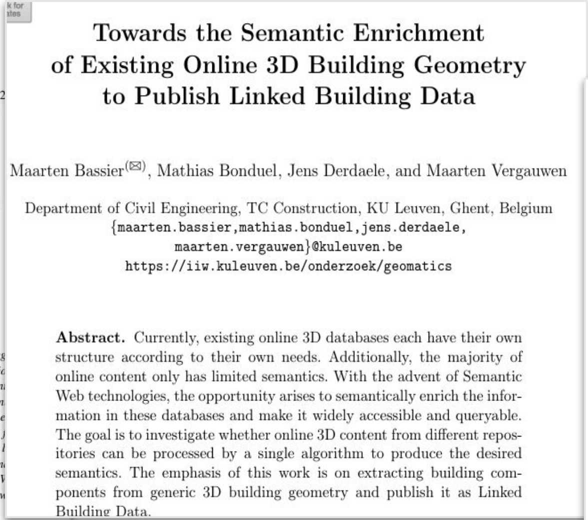
Three-dimensional information exchange over the semantic web for the domain of architecture, engineering, and construction

PIETER PAUWELS,[1,2] DAVY VAN DEURSEN,[3] JOS DE ROO,[3] TIM VAN ACKERE,[1]
RONALD DE MEYER,[1] RIK VAN DE WALLE,[3] AND JAN VAN CAMPENHOUT[2]
[1]Department of Architecture and Urban Planning, Ghent University, Ghent, Belgium
[2]Department of Electronics and Information Systems, Ghent University, Ghent, Belgium
[3]Department of Electronics and Information Systems, Multimedia Lab, Ghent University, IBBT, Ledeberg-Ghent, Belgium

(RECEIVED September 15, 2010; ACCEPTED February 7, 2011)

Abstract

Three-dimensional (3-D) geometry can be described in many ways, with both a varying syntax and a varying semantics. As a result, several very diverse schemas and file formats can be deployed to describe geometry, depending on the application domain in question. In a multidisciplinary domain such as the domain of architecture, engineering, and construction, this range of specialized schemas makes file format conversions inevitable. The approach adopted by current conversion tools, however, often results in a loss of information, most often due to a "mistranslation" between different syntaxes and/or semantics, leading to errors and limitations in the design conception stage and to inefficiency due to the required remodeling efforts. An approach based on semantic web technology may reduce the loss of information significantly, leading to an improved processing of 3-D information and hence to an improved design practice in the architecture, engineering, and construction domain. This paper documents our investigation of the nature of this 3-D information conversion problem and how it may be encompassed using semantic web technology. In an exploratory double test case, we show how the specific deployment of semantic rule languages and an appropriate inference engine are to be adopted to improve this 3-D information exchange. It shows how semantic web technology allows the coexistence of diverse descriptions of the same 3-D information, interlinked through explicit conversion rules. Although only a simple example is used to document the process, and a more in-depth investigation is needed, the initial results indicate the suggested approach to be a useful alternative approach

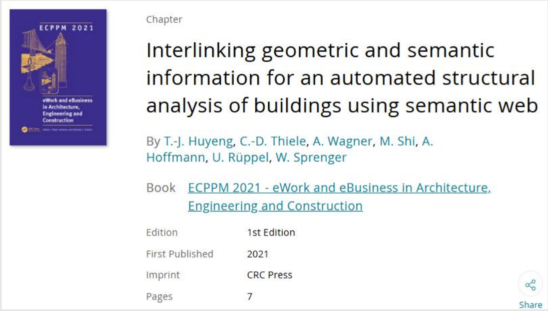


Towards the Semantic Enrichment
of Existing Online 3D Building Geometry
to Publish Linked Building Data

Maarten Bassier(✉), Mathias Bonduel, Jens Derdaele, and Maarten Vergauwen

Department of Civil Engineering, TC Construction, KU Leuven, Ghent, Belgium
{maarten.bassier,mathias.bonduel,jens.derdaele,
maarten.vergauwen}@kuleuven.be
https://iiw.kuleuven.be/onderzoek/geomatics

Abstract. Currently, existing online 3D databases each have their own structure according to their own needs. Additionally, the majority of online content only has limited semantics. With the advent of Semantic Web technologies, the opportunity arises to semantically enrich the information in these databases and make it widely accessible and queryable. The goal is to investigate whether online 3D content from different repositories can be processed by a single algorithm to produce the desired semantics. The emphasis of this work is on extracting building components from generic 3D building geometry and publish it as Linked Building Data.



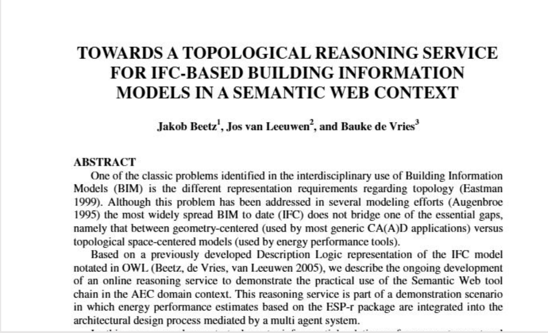
Chapter

Interlinking geometric and semantic information for an automated structural analysis of buildings using semantic web

By T.-J. Huyeng, C.-D. Thiele, A. Wagner, M. Shi, A. Hoffmann, U. Rüppel, W. Sprenger

Book ECPPM 2021 - eWork and eBusiness in Architecture, Engineering and Construction

Edition 1st Edition
First Published 2021
Imprint CRC Press
Pages 7



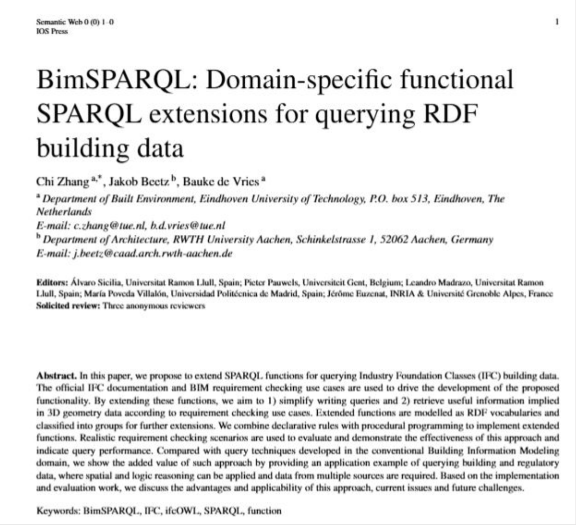
TOWARDS A TOPOLOGICAL REASONING SERVICE
FOR IFC-BASED BUILDING INFORMATION
MODELS IN A SEMANTIC WEB CONTEXT

Jakob Beetz[1], Jos van Leeuwen[2], and Bauke de Vries[3]

ABSTRACT

One of the classic problems identified in the interdisciplinary use of Building Information Models (BIM) is the different representation requirements regarding topology (Eastman 1999). Although this problem has been addressed in several modeling efforts (Augenbroe 1995) the most widely spread BIM to date (IFC) does not bridge one of the essential gaps, namely that between geometry-centered (used by most generic CA(A)D applications) versus topological space-centered models (used by energy performance tools).

Based on a previously developed Description Logic representation of the IFC model notated in OWL (Beetz, de Vries, van Leeuwen 2005), we describe the ongoing development of an online reasoning service to demonstrate the practical use of the Semantic Web tool chain in the AEC domain context. This reasoning service is part of a demonstration scenario in which energy performance estimates based on the ESP-r package are integrated into the architectural design process mediated by a multi agent system.



Semantic Web 0 (0) 1–0
IOS Press

1

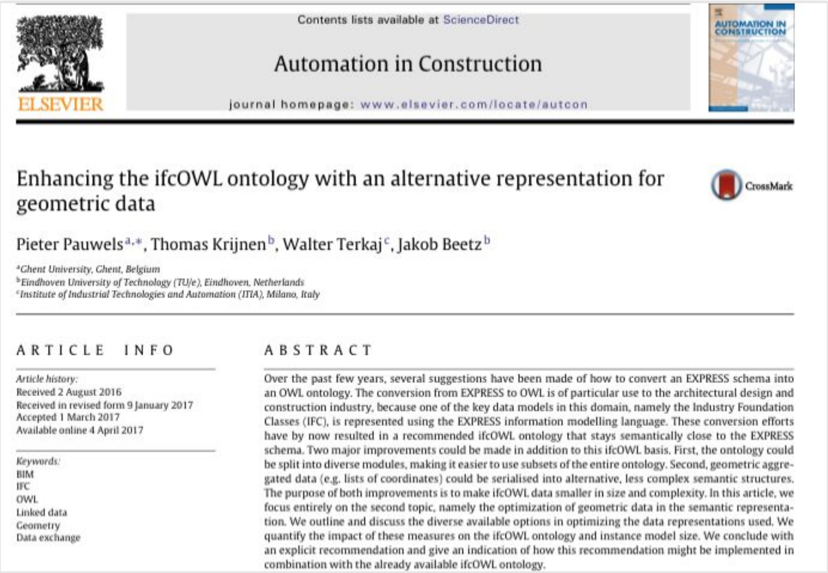
BimSPARQL: Domain-specific functional SPARQL extensions for querying RDF building data

Chi Zhang[a,*], Jakob Beetz[b], Bauke de Vries[a]
[a] *Department of Built Environment, Eindhoven University of Technology, P.O. box 513, Eindhoven, The Netherlands*
E-mail: c.zhang@tue.nl, b.d.vries@tue.nl
[b] *Department of Architecture, RWTH University Aachen, Schinkelstrasse 1, 52062 Aachen, Germany*
E-mail: j.beetz@caad.arch.rwth-aachen.de

Editors: Álvaro Sicilia, Universitat Ramon Llull, Spain; Pieter Pauwels, Universiteit Gent, Belgium; Leandro Madrazo, Universitat Ramon Llull, Spain; María Poveda Villalón, Universidad Politécnica de Madrid, Spain; Jérôme Euzenat, INRIA & Université Grenoble Alpes, France
Solicited review: Three anonymous reviewers

Abstract. In this paper, we propose to extend SPARQL functions for querying Industry Foundation Classes (IFC) building data. The official IFC documentation and BIM requirement checking use cases are used to drive the development of the proposed functionality. By extending these functions, we aim to 1) simplify writing queries and 2) retrieve useful information implied in 3D geometry data according to requirement checking use cases. Extended functions are modelled as RDF vocabularies and classified into groups for further extensions. We combine declarative rules with procedural programming to implement extended functions. Realistic requirement checking scenarios are used to evaluate and demonstrate the effectiveness of this approach and indicate query performance. Compared with query techniques developed in the conventional Building Information Modeling domain, we show the added value of such approach by providing an application example of querying building and regulatory data, where spatial and logic reasoning can be applied and data from multiple sources are required. Based on the implementation and evaluation work, we discuss the advantages and applicability of this approach, current issues and future challenges.

Keywords: BimSPARQL, IFC, ifcOWL, SPARQL, function



Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Enhancing the ifcOWL ontology with an alternative representation for geometric data

Pieter Pauwels[a,*], Thomas Krijnen[b], Walter Terkaj[c], Jakob Beetz[b]
[a]*Ghent University, Ghent, Belgium*
[b]*Eindhoven University of Technology (TU/e), Eindhoven, Netherlands*
[c]*Institute of Industrial Technologies and Automation (ITIA), Milano, Italy*

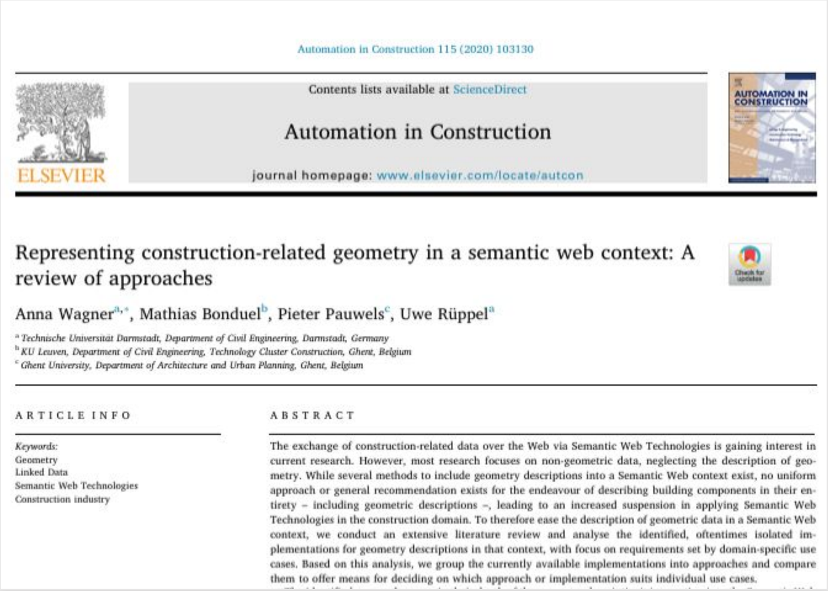
ARTICLE INFO

Article history:
Received 2 August 2016
Received in revised form 9 January 2017
Accepted 1 March 2017
Available online 4 April 2017

Keywords:
BIM
IFC
OWL
Linked data
Geometry
Data exchange

ABSTRACT

Over the past few years, several suggestions have been made of how to convert an EXPRESS schema into an OWL ontology. The conversion from EXPRESS to OWL is of particular use to the architectural design and construction industry, because one of the key data models in this domain, namely the Industry Foundation Classes (IFC), is represented using the EXPRESS information modelling language. These conversion efforts have by now resulted in a recommended ifcOWL ontology that stays semantically close to the EXPRESS schema. Two major improvements could be made in addition to this ifcOWL basis. First, the ontology could be split into diverse modules, making it easier to use subsets of the entire ontology. Second, geometric aggregated data (e.g. lists of coordinates) could be serialised into alternative, less complex semantic structures. The purpose of both improvements is to make ifcOWL data smaller in size and complexity. In this article, we focus entirely on the second topic, namely the optimization of geometric data in the semantic representation. We outline and discuss the diverse available options in optimizing the data representations used. We quantify the impact of these measures on the ifcOWL ontology and instance model size. We conclude with an explicit recommendation and give an indication of how this recommendation might be implemented in combination with the already available ifcOWL ontology.



Automation in Construction 115 (2020) 103130

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Representing construction-related geometry in a semantic web context: A review of approaches

Anna Wagner[a,*], Mathias Bonduel[b], Pieter Pauwels[c], Uwe Rüppel[a]
[a] *Technische Universität Darmstadt, Department of Civil Engineering, Darmstadt, Germany*
[b] *KU Leuven, Department of Civil Engineering, Technology Cluster Construction, Ghent, Belgium*
[c] *Ghent University, Department of Architecture and Urban Planning, Ghent, Belgium*

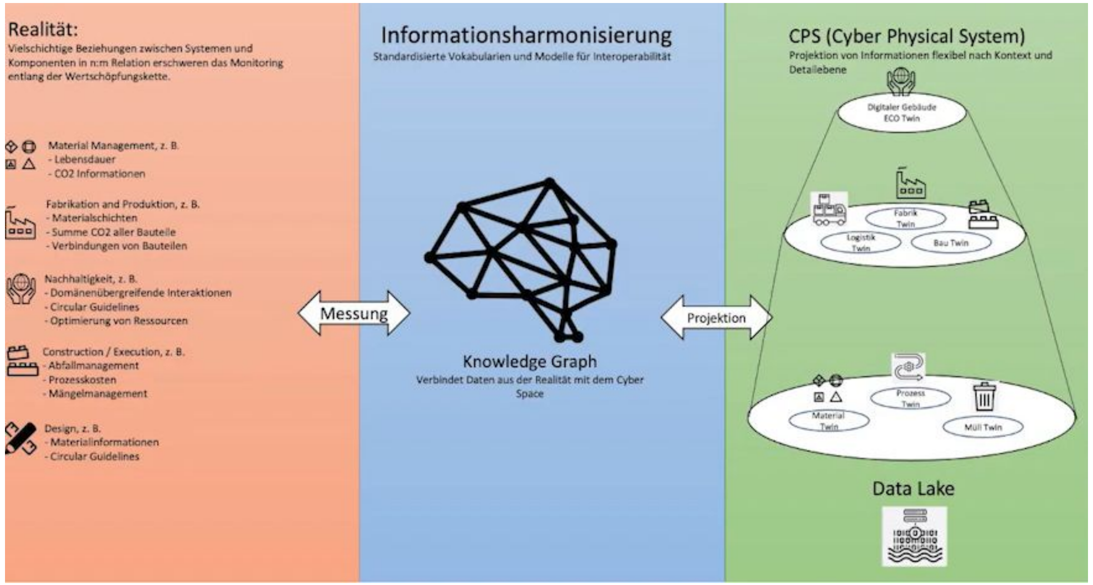
ARTICLE INFO

Keywords:
Geometry
Linked Data
Semantic Web Technologies
Construction industry

ABSTRACT

The exchange of construction-related data over the Web via Semantic Web Technologies is gaining interest in current research. However, most research focuses on non-geometric data, neglecting the description of geometry. While several methods to include geometry descriptions into a Semantic Web context exist, no uniform approach or general recommendation exists for the endeavour of describing building components in their entirety – including geometric descriptions –, leading to an increased suspension in applying Semantic Web Technologies in the construction domain. To therefore ease the description of geometric data in a Semantic Web context, we conduct an extensive literature review and analyse the identified, oftentimes isolated implementations for geometry descriptions in that context, with focus on requirements set by domain-specific use cases. Based on this analysis, we group the currently available implementations into approaches and compare them to offer means for deciding on which approach or implementation suits individual use cases.

3D Geometry and Semantic Web - Industry Context: GROPYUS



Realität:
Vielschichtige Beziehungen zwischen Systemen und Komponenten in n:m Relation erschweren das Monitoring entlang der Wertschöpfungskette.

Material Management, z. B.
- Lebensdauer
- CO2 Informationen

Fabrikation and Produktion, z. B.
- Materialschichten
- Summe CO2 aller Bauteile
- Verbindungen von Bauteilen

Nachhaltigkeit, z. B.
- Domänenübergreifende Interaktionen
- Circular Guidelines
- Optimierung von Ressourcen

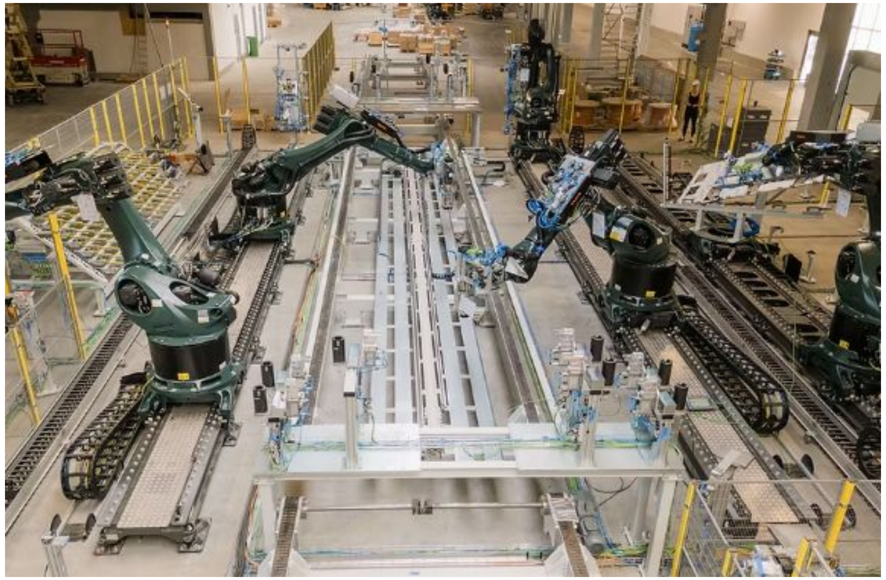
Construction / Execution, z. B.
- Abfallmanagement
- Prozesskosten
- Mängelmanagement

Design, z. B.
- Materialinformationen
- Circular Guidelines

Informationsharmonisierung
Standardisierte Vokabularien und Modelle für Interoperabilität

Knowledge Graph
Verbindet Daten aus der Realität mit dem Cyber Space

CPS (Cyber Physical System)
Projektion von Informationen flexibel nach Kontext und Detailebene

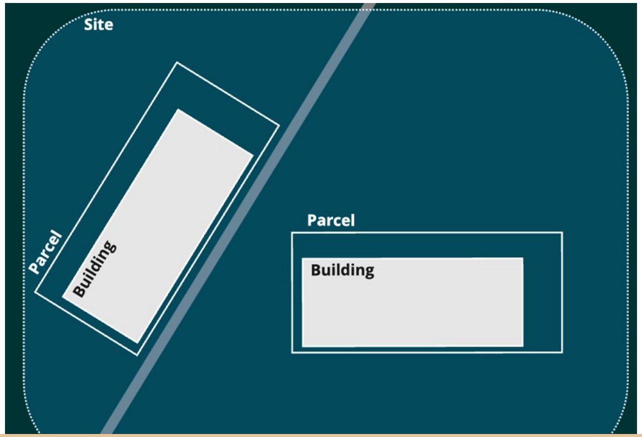


Who/What is Gropyus?
- Construction company / general contractor
- Offers digital, end-to-end solution — from design and production to assembly and smart building operations,
- 350–440 staff across Europe.

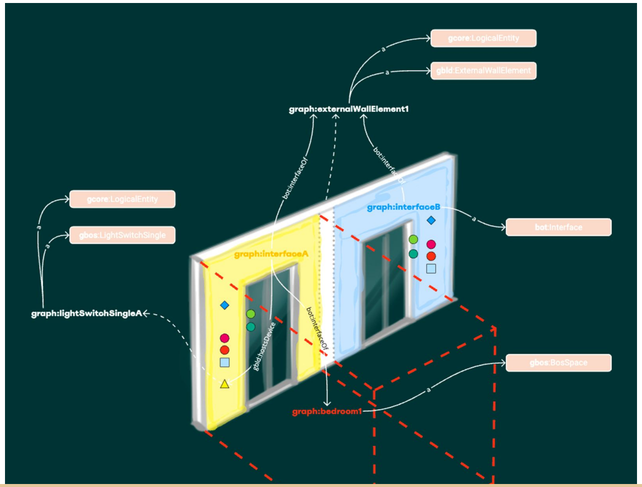
- Timber Building Design
- Robotic Fabrication
- **Use knowledge graphs for data representation and integration** (using a data fabric architecture)

Bossong, E. (2022, December 6). NetworkMeeting – Erik Bossong about Eco-Twin [Video]. Platform Digitize Wood. YouTube. https://youtu.be/AJg_KZADb6I?t=1451
Slide image retrieved from minute 15:32, "Modellentwicklung Eco-Twin," GROPYUS Company Presentation, March 2022.

3D Geometry and Semantic Web - Industry Context: GROPYUS



Problem: Transformations/ Projections



Problem: Geometric Associations Wall interfaces with Rooms and Spaces

- **No explicit room modeling**: spatial queries like *"which room contains this device?"* were not possible.
- **No geometric reasoning**: relationships like containment or adjacency could not be inferred from geometry.
- **Ad-hoc geometry formats**: custom representations limited interoperability and reuse.
- **No native support for transformations:** geometric placements couldn't be resolved directly in RDF.
- **No derived computations**: basic metrics like area or volume couldn't be calculated or validated.

Requirements:
R1: Support for multiple geometry types (BRep, CSG).
R2: Validation of geometry.
R3: Spatial operations (e.g., volume, containment).
R4: Scalable modeling across many units (serial production).

Storing Geometry in Databases

Encoding: Geometric Data

Literals (WKT GML) & Ontologies (OntoBREP)

Describing: Geometric Metadata

Relationships & Metadata (OMG, FOG)

Querying: Geometric Operations

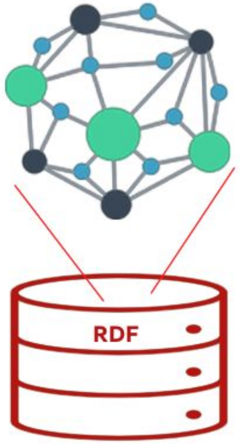
Spatial reasoning and queries (GeoSPARQL, BimSPARQL)

Comparative Analysis of Approaches for Geometric Data Representation in RDF. by Diellza Elshani, Ali Nakhaee, Anthony A. Arrascue, Haris Isakovic, Navid Hedayati, Janakiram Karlapudi, Thomas Wortmann
LDAC 2025

Approaches to Representing and Querying Geometry in RDF

Approaches to Representing and Querying Geometry in RDF

Approach I:
Represent geometry using RDF (using an ontology or not) but have no dedicated query language for that.

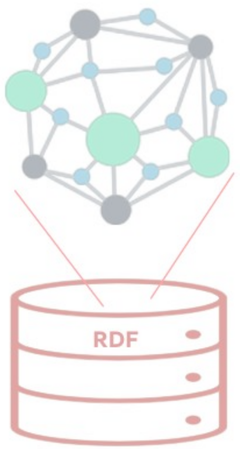


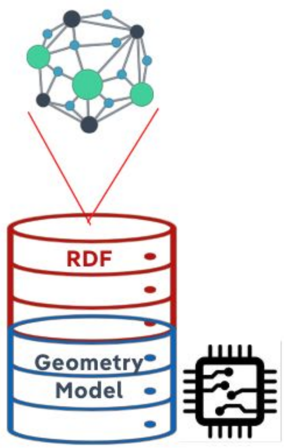
Comparative Analysis of Approaches for Geometric Data Representation in RDF. by Diellza Elshani, Ali Nakhaee, Anthony A. Arrascue, Haris Isakovic, Navid Hedayati, Janakiram Karlapudi, Thomas Wortmann LDAC 2025

Approaches to Representing and Querying Geometry in RDF

Approach I:
Represent geometry using RDF (using an ontology or not) but have no dedicated query language for that.

Approach 2: Geometry data is represented in RDF/OWL and can be queried using a dedicated query language within the RDF database, such as GeoSPARQL or BimSPARQL.





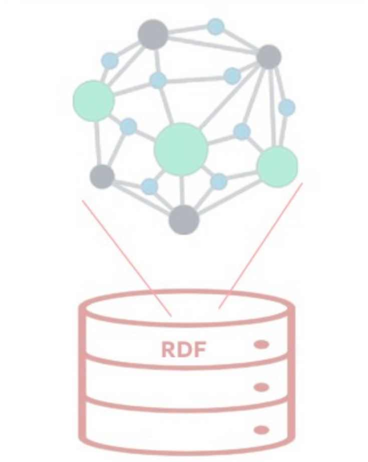
Comparative Analysis of Approaches for Geometric Data Representation in RDF. by Diellza Elshani, Ali Nakhaee, Anthony A. Arrascue, Haris Isakovic, Navid Hedayati, Janakiram Karlapudi, Thomas Wortmann LDAC 2025

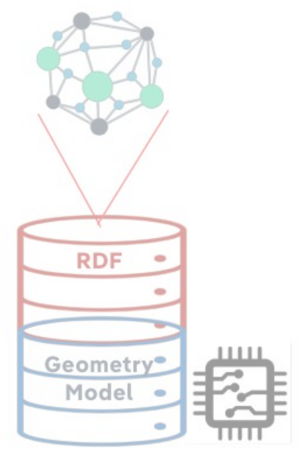
Approaches to Representing and Querying Geometry in RDF

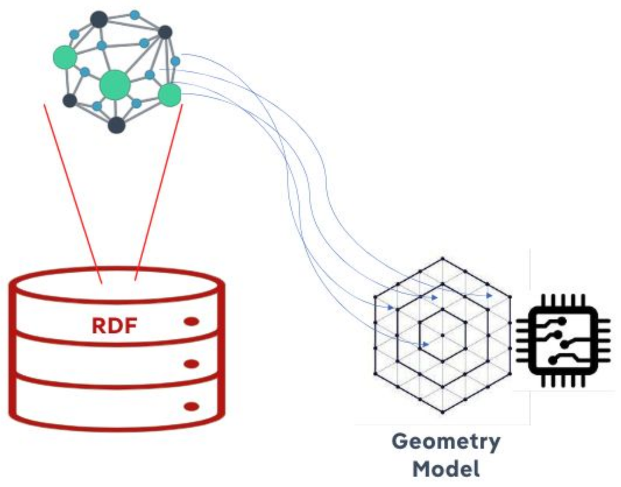
Approach I:
Represent geometry using RDF (using an ontology or not) but have no dedicated query language for that.

Approach 2: Geometry data is represented in RDF/OWL and can be queried using a dedicated query language within the RDF database, such as GeoSPARQL or BimSPARQL.

Approach 3: Geometry is linked from RDF to external geometry engines or spatial databases, which handle computation and validation outside the RDF store.

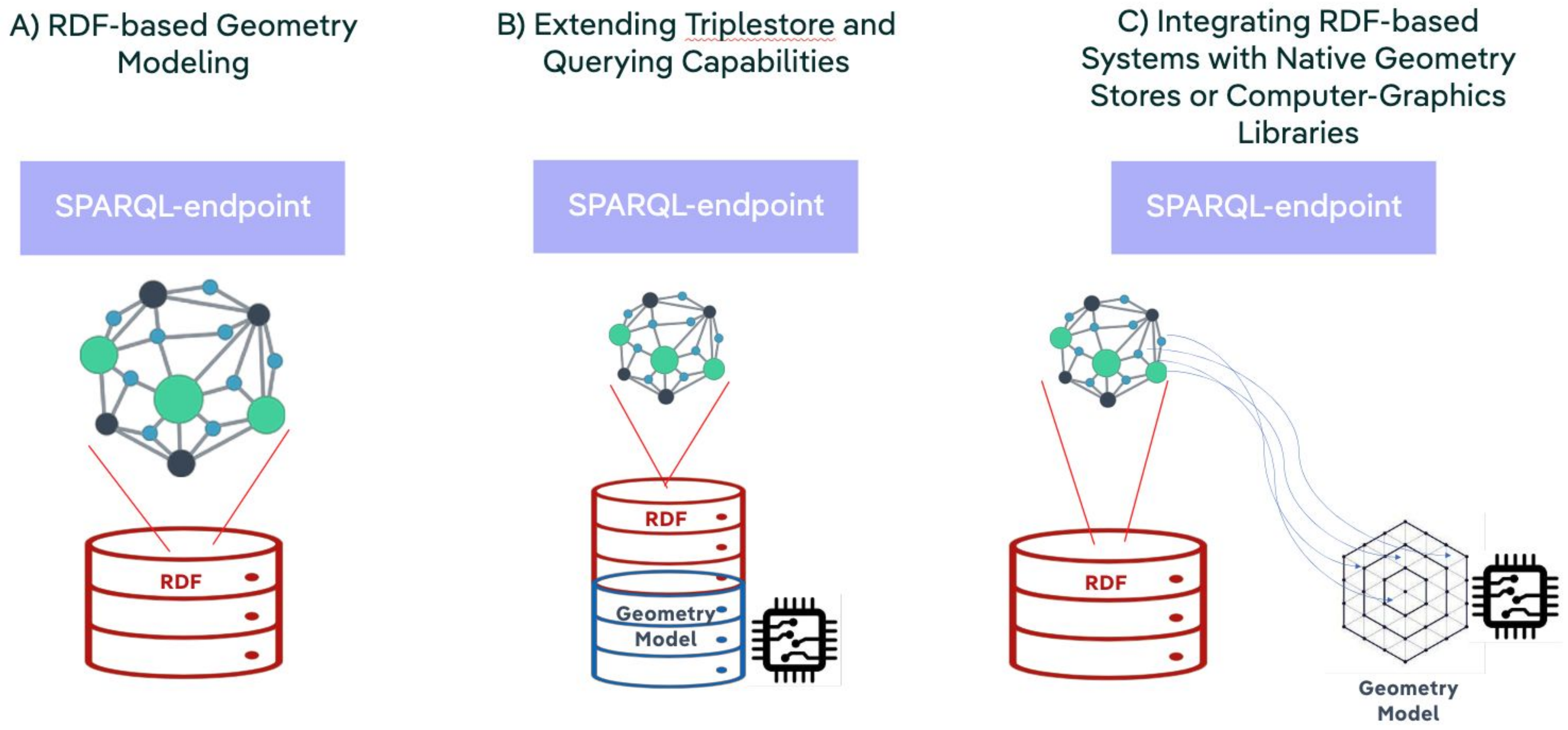






Comparative Analysis of Approaches for Geometric Data Representation in RDF. by Diellza Elshani, Ali Nakhaee, Anthony A. Arrascue, Haris Isakovic, Navid Hedayati, Janakiram Karlapudi, Thomas Wortmann LDAC 2025

Approaches to Representing and Querying Geometry in RDF



A) RDF-based Geometry Modeling

B) Extending Triplestore and Querying Capabilities

C) Integrating RDF-based Systems with Native Geometry Stores or Computer-Graphics Libraries

Comparative Analysis of Approaches for Geometric Data Representation in RDF. by Diellza Elshani, Ali Nakhaee, Anthony A. Arrascue, Haris Isakovic, Navid Hedayati, Janakiram Karlapudi, Thomas Wortmann LDAC 2025

Alternative categorizations of RDF geometry exist—such as those based on data structure or encoding—but they do not address geometry querying, processing, or scalability, which are the focus of our work.

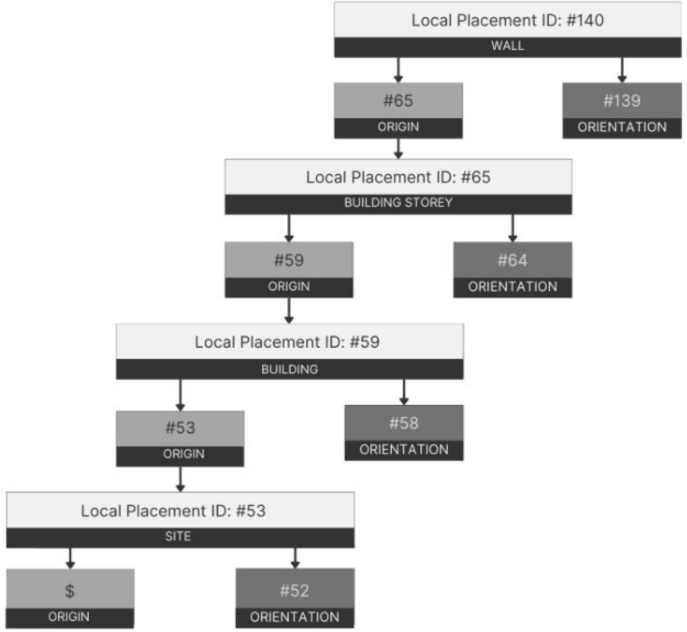
Criteria Used for Evaluation:

Each approach was assessed across the following dimensions:

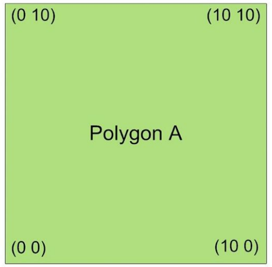
- **Expressiveness** (ability to represent complex geometry and relationships)
- **Complexity** (ease of use and implementation)
- **Geometry Support** (how geometry is stored/represented)
- **Geometric Functions** (what spatial operations are possible)
- **Geometry Validation** (support for checking geometric correctness)
- **Scalability** (performance with large datasets)
- **Interoperability** (ability to work with external systems)
- **Standardization** (alignment with industry standards)

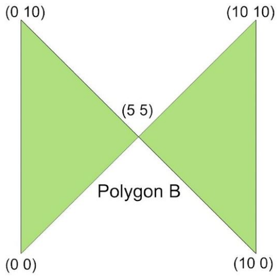
Range: High, medium, low, none

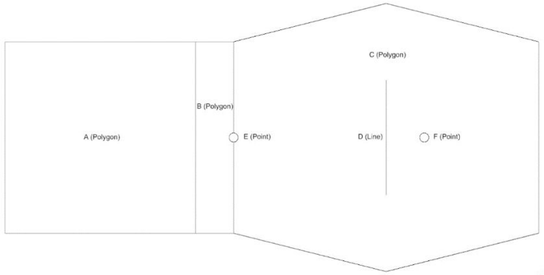
Dataset and Systems Involved

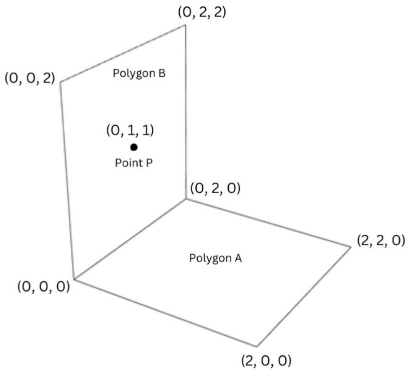

Local Placement ID: #140
WALL
#65
ORIGIN
#139
ORIENTATION
Local Placement ID: #65
BUILDING STOREY
#59
ORIGIN
#64
ORIENTATION
Local Placement ID: #59
BUILDING
#53
ORIGIN
#58
ORIENTATION
Local Placement ID: #53
SITE
$
ORIGIN
#52
ORIENTATION

- Used a simplified IFC file as a base model for geometry testing
- Model includes walls, spaces, devices, and transformation data
- Designed and ran custom unit tests on geometric primitives
- Tested operations like containment, validation, CSG, and transformations
- Evaluated behavior across multiple tools (e.g., PostGIS, CGAL, Trimesh)





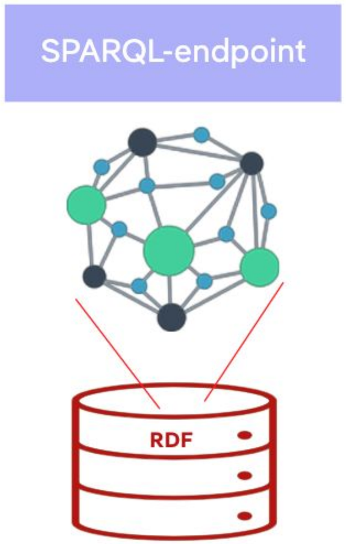


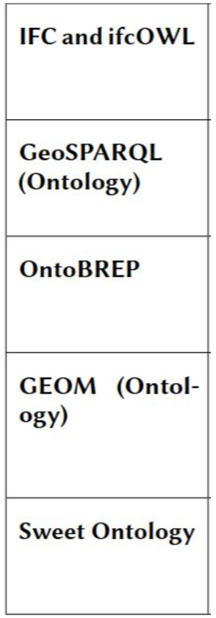


Comparative Analysis of Approaches for Geometric Data Representation in RDF. by Diellza Elshani, Ali Nakhaee, Anthony A. Arrascue, Haris Isakovic, Navid Hedayati, Janakiram Karlapudi, Thomas Wortmann LDAC 2025

Approach I: RDF-based geometry Modelling

A) RDF-based Geometry Modeling





IFC and ifcOWL
GeoSPARQL (Ontology)
OntoBREP
GEOM (Ontology)
Sweet Ontology

Comparative Analysis of Approaches for Geometric Data Representation in RDF. by Diellza Elshani, Ali Nakhaee, Anthony A. Arrascue, Haris Isakovic, Navid Hedayati, Janakiram Karlapudi, Thomas Wortmann LDAC 2025

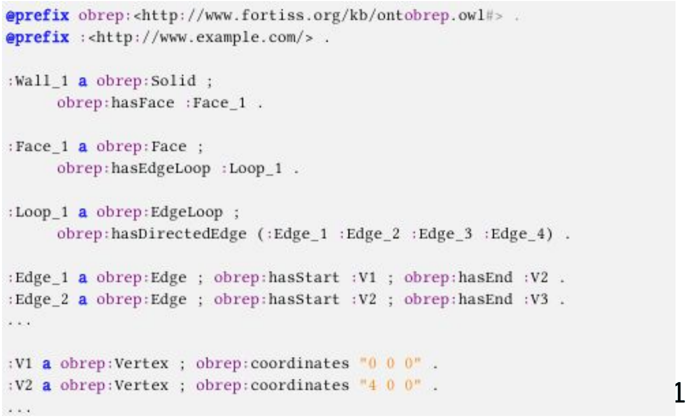
ifcOWLRepresentation(CSG-based)

```
@prefix :<http://www.example.com/> .
@prefix ifc:<https://standards.buildingsmart.org/IFC/DEV/IFC4/ADD2_TC1/OWL#> .

:Wall_1 a ifc:IfcWallStandardCase ;
     ifc:Representation :WallGeom_1 .

:WallGeom_1 a ifc:IfcProductDefinitionShape ;
        ifc:Representations :ShapeRep_1 .

:ShapeRep_1 a ifc:IfcShapeRepresentation ;
        ifc:Items :Extrusion_1 .

:Extrusion_1 a ifc:IfcExtrudedAreaSolid ;
         ifc:SweptArea :RectProfile_1 ;
         ifc:Depth "0.2"^^xsd:double ;
         ifc:ExtrudedDirection :ZDir .

:RectProfile_1 a ifc:IfcRectangleProfileDef ;
          ifc:XDim "4.0"^^xsd:double ;
          ifc:YDim "3.0"^^xsd:double .
```

GeoSPARQL Representation (WKT-based)

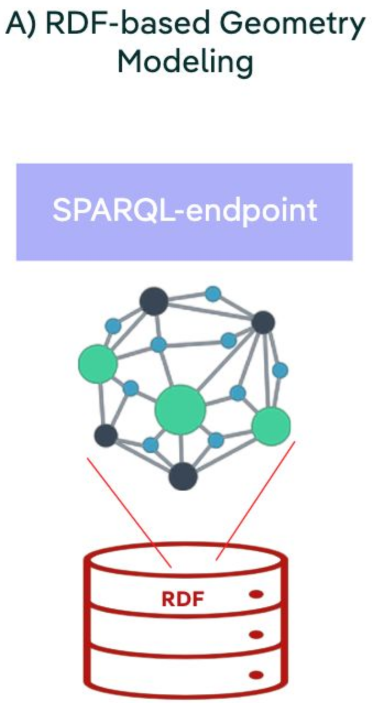
```
@prefix geo:<http://www.opengis.net/ont/geosparql#> .
@prefix :<http://www.example.com/> .

:Wall_1 a geo:Feature ;
     geo:hasGeometry :WallGeom_1 .

:WallGeom_1 a geo:Geometry ;
        geo:asWKT "POLYGON((0 0,4 0,4 0.2,0 0.2,0 0))"^^geo:wktLiteral ;
        geo:crs <http://www.opengis.net/def/crs/EPSG/0/4326> .
```

OntoBREP Representation (Topological structure)

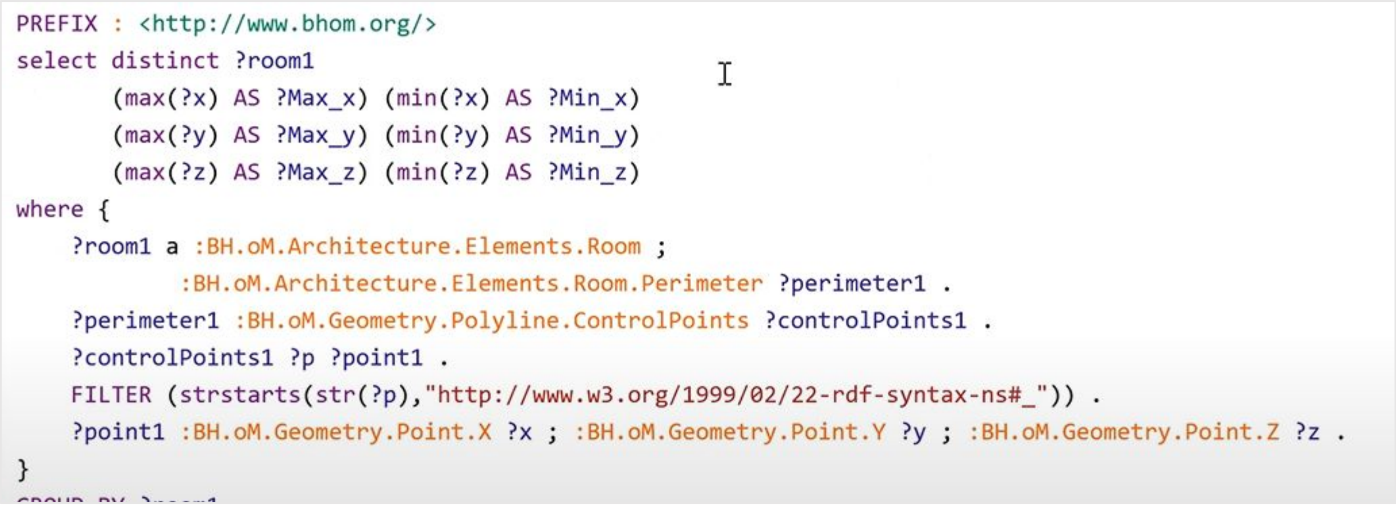


```
@prefix obrep:<http://www.fortiss.org/kb/ontobrep.owl#> .
@prefix :<http://www.example.com/> .

:Wall_1 a obrep:Solid ;
      obrep:hasFace :Face_1 .

:Face_1 a obrep:Face ;
      obrep:hasEdgeLoop :Loop_1 .

:Loop_1 a obrep:EdgeLoop ;
      obrep:hasDirectedEdge (:Edge_1 :Edge_2 :Edge_3 :Edge_4) .

:Edge_1 a obrep:Edge ; obrep:hasStart :V1 ; obrep:hasEnd :V2 .
:Edge_2 a obrep:Edge ; obrep:hasStart :V2 ; obrep:hasEnd :V3 .
...

:V1 a obrep:Vertex ; obrep:coordinates "0 0 0" .
:V2 a obrep:Vertex ; obrep:coordinates "4 0 0" .
...
```

Approach I: RDF-based geometry Modelling



IFC and ifcOWL
GeoSPARQL (Ontology)
OntoBREP
GEOM (Ontology)
Sweet Ontology

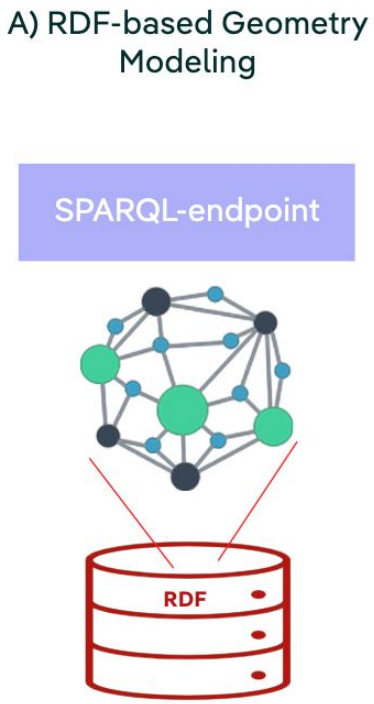


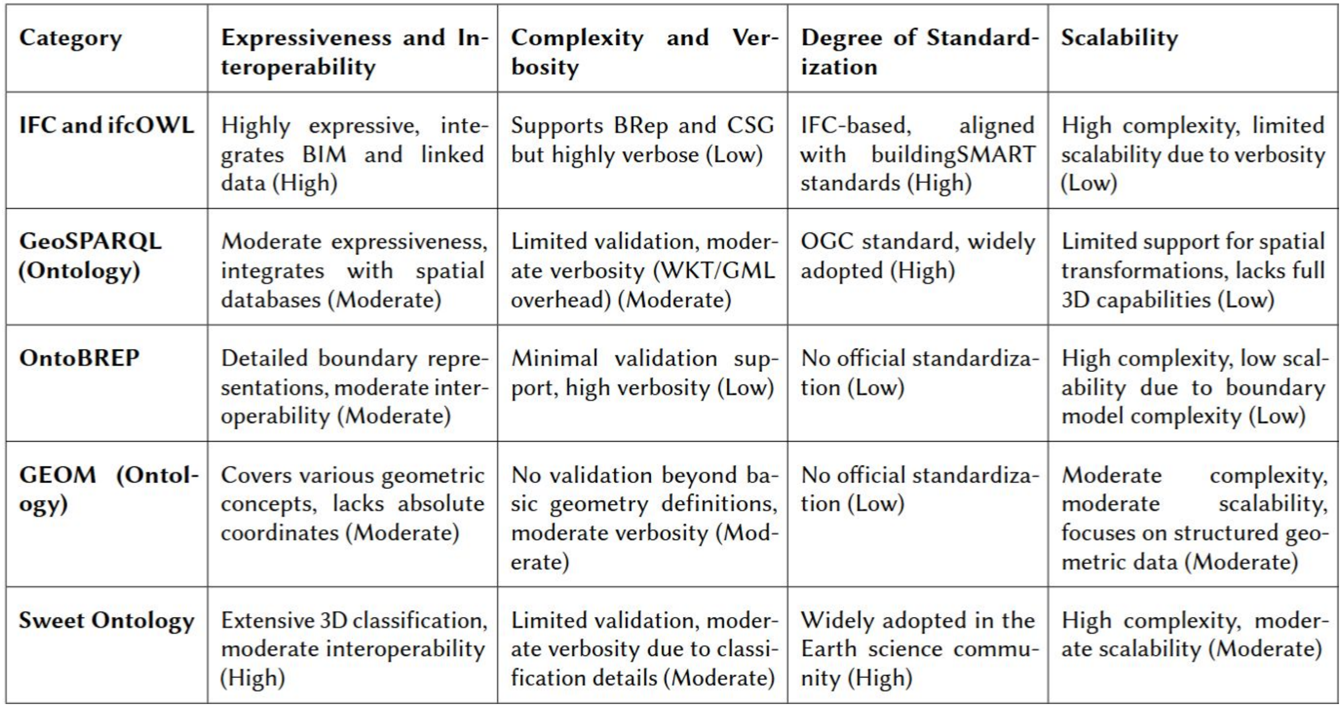
```
PREFIX : <http://www.bhom.org/>
select distinct ?room1
       (max(?x) AS ?Max_x) (min(?x) AS ?Min_x)
       (max(?y) AS ?Max_y) (min(?y) AS ?Min_y)
       (max(?z) AS ?Max_z) (min(?z) AS ?Min_z)
where {
    ?room1 a :BH.oM.Architecture.Elements.Room ;
            :BH.oM.Architecture.Elements.Room.Perimeter ?perimeter1 .
    ?perimeter1 :BH.oM.Geometry.Polyline.ControlPoints ?controlPoints1 .
    ?controlPoints1 ?p ?point1 .
    FILTER (strstarts(str(?p),"http://www.w3.org/1999/02/22-rdf-syntax-ns#_")) .
    ?point1 :BH.oM.Geometry.Point.X ?x ; :BH.oM.Geometry.Point.Y ?y ; :BH.oM.Geometry.Point.Z ?z .
}
```

```
SELECT DISTINCT ?room1 ?room2
WHERE {
    {
        select distinct ?room1 ?room2
               (max(?x1) AS ?Max_x1) (min(?x1) AS ?Min_x1)
               (max(?y1) AS ?Max_y1) (min(?y1) AS ?Min_y1)
               (max(?z1) AS ?Max_z1) (min(?z1) AS ?Min_z1)
               (max(?x2) AS ?Max_x2) (min(?x2) AS ?Min_x2)
               (max(?y2) AS ?Max_y2) (min(?y2) AS ?Min_y2)
               (max(?z2) AS ?Max_z2) (min(?z2) AS ?Min_z2)
        where {
            ?room1 a :BH.oM.Architecture.Elements.Room ;
                    :BH.oM.Architecture.Elements.Room.Perimeter ?perimeter1 .
```

Comparative Analysis of Approaches for Geometric Data Representation in RDF. by Diellza Elshani, Ali Nakhaee, Anthony A. Arrascue, Haris Isakovic, Navid Hedayati, Janakiram Karlapudi, Thomas Wortmann LDAC 2025

Approach I: RDF-based geometry Modelling


A) RDF-based Geometry Modeling

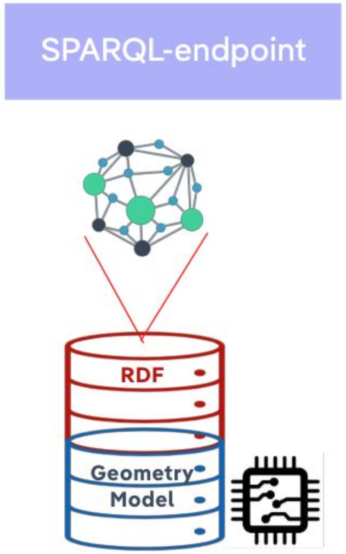


Category	Expressiveness and Interoperability	Complexity and Verbosity	Degree of Standardization	Scalability
IFC and ifcOWL	Highly expressive, integrates BIM and linked data (High)	Supports BRep and CSG but highly verbose (Low)	IFC-based, aligned with buildingSMART standards (High)	High complexity, limited scalability due to verbosity (Low)
GeoSPARQL (Ontology)	Moderate expressiveness, integrates with spatial databases (Moderate)	Limited validation, moderate verbosity (WKT/GML overhead) (Moderate)	OGC standard, widely adopted (High)	Limited support for spatial transformations, lacks full 3D capabilities (Low)
OntoBREP	Detailed boundary representations, moderate interoperability (Moderate)	Minimal validation support, high verbosity (Low)	No official standardization (Low)	High complexity, low scalability due to boundary model complexity (Low)
GEOM (Ontology)	Covers various geometric concepts, lacks absolute coordinates (Moderate)	No validation beyond basic geometry definitions, moderate verbosity (Moderate)	No official standardization (Low)	Moderate complexity, moderate scalability, focuses on structured geometric data (Moderate)
Sweet Ontology	Extensive 3D classification, moderate interoperability (High)	Limited validation, moderate verbosity due to classification details (Moderate)	Widely adopted in the Earth science community (High)	High complexity, moderate scalability (Moderate)

Comparative Analysis of Approaches for Geometric Data Representation in RDF. by Diellza Elshani, Ali Nakhaee, Anthony A. Arrascue, Haris Isakovic, Navid Hedayati, Janakiram Karlapudi, Thomas Wortmann LDAC 2025

Approach II: Extending Triplestore and Querying Capabilities

B) Extending Triplestore and Querying Capabilities



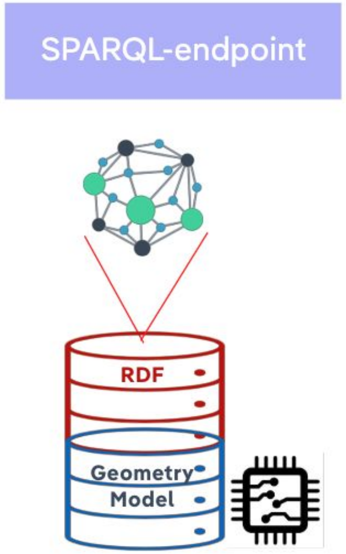
GeoSPARQL (Query Language)

BimSPARQL

Comparative Analysis of Approaches for Geometric Data Representation in RDF. by Diellza Elshani, Ali Nakhaee, Anthony A. Arrascue, Haris Isakovic, Navid Hedayati, Janakiram Karlapudi, Thomas Wortmann LDAC 2025

Approach II: Extending Triplestore and Querying Capabilities

B) Extending Triplestore and Querying Capabilities



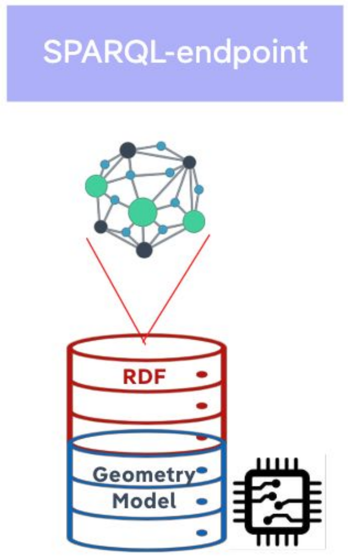
GeoSPARQL (Query Language)
BimSPARQL

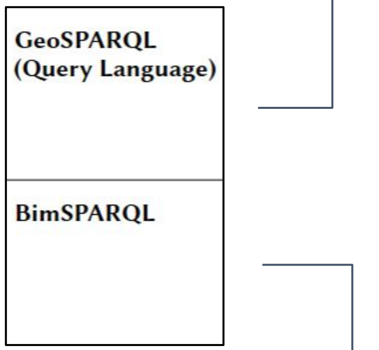
Dedicated query languages enable efficient, domain-specific geometry processing and semantic inference within RDF systems.

Comparative Analysis of Approaches for Geometric Data Representation in RDF. by Diellza Elshani, Ali Nakhaee, Anthony A. Arrascue, Haris Isakovic, Navid Hedayati, Janakiram Karlapudi, Thomas Wortmann LDAC 2025

Approach II: Extending Triplestore and Querying Capabilities

B) Extending Triplestore and Querying Capabilities





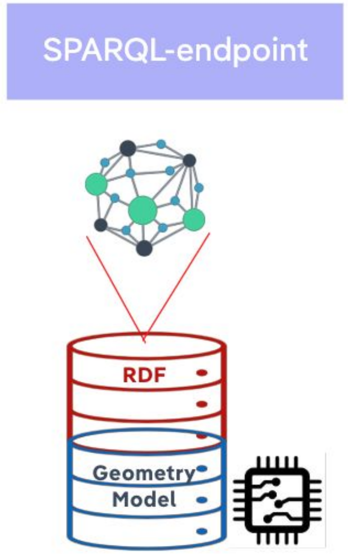
- Enables standardized 2D spatial operations (e.g., intersects, within, distance) on RDF data, making it effective for geospatial querying and reasoning.
- Supports geometry literals in WKT (geo:wktLiteral) and GML (geo:gmlLiteral) WKT is most commonly used due to simplicity and broad tool support
- Limited 3D geometry support
- No built-in geometry validation.

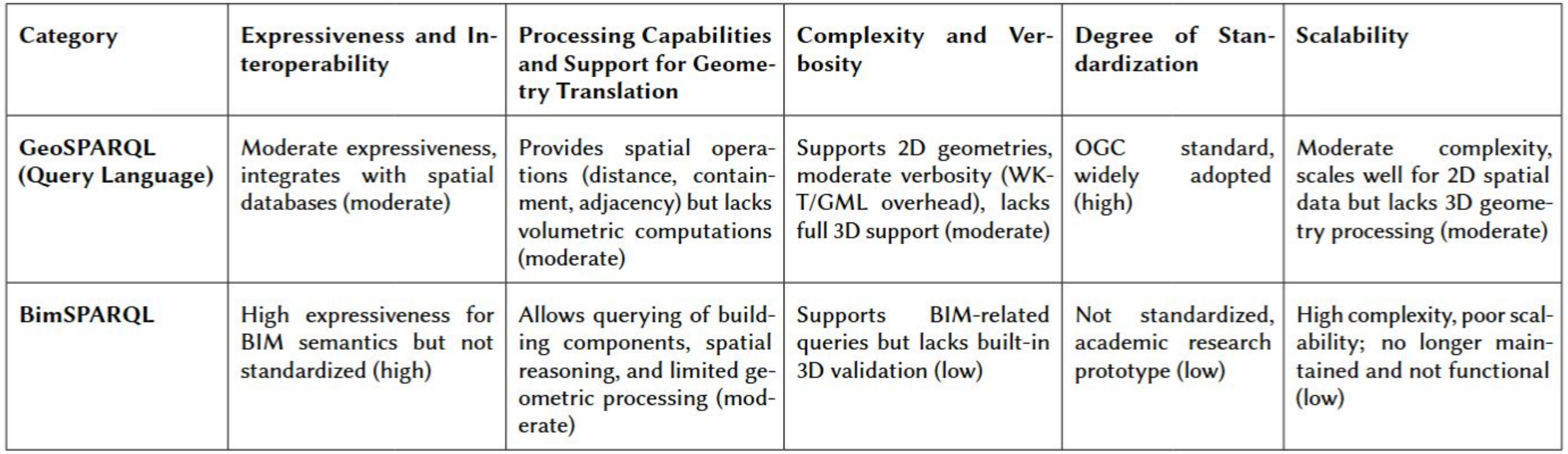
- Supports 3D spatial operations (e.g., intersects, within, distance) aligned with OGC standards.
- Accepts WKT literals for spatial operations, aligned with GeoSPARQL.
- Difficulties to test: outdated and dependent on deprecated SPIN (which is no longer maintained and has been superseded by SHACL, the W3C recommendation for expressing rules, constraints, and inference in RDF).
- Documented many 3D functions, but missing ones needed for GROPYUS use case.

Comparative Analysis of Approaches for Geometric Data Representation in RDF. by Diellza Elshani, Ali Nakhaee, Anthony A. Arrascue, Haris Isakovic, Navid Hedayati, Janakiram Karlapudi, Thomas Wortmann LDAC 2025

Approach II: Extending Triplestore and Querying Capabilities

B) Extending Triplestore and Querying Capabilities





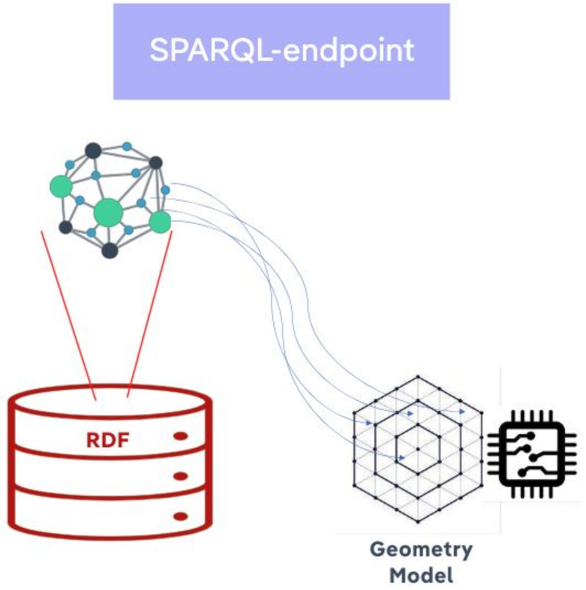
Category	Expressiveness and Interoperability	Processing Capabilities and Support for Geometry Translation	Complexity and Verbosity	Degree of Standardization	Scalability
GeoSPARQL (Query Language)	Moderate expressiveness, integrates with spatial databases (moderate)	Provides spatial operations (distance, containment, adjacency) but lacks volumetric computations (moderate)	Supports 2D geometries, moderate verbosity (WKT/GML overhead), lacks full 3D support (moderate)	OGC standard, widely adopted (high)	Moderate complexity, scales well for 2D spatial data but lacks 3D geometry processing (moderate)
BimSPARQL	High expressiveness for BIM semantics but not standardized (high)	Allows querying of building components, spatial reasoning, and limited geometric processing (moderate)	Supports BIM-related queries but lacks built-in 3D validation (low)	Not standardized, academic research prototype (low)	High complexity, poor scalability; no longer maintained and not functional (low)

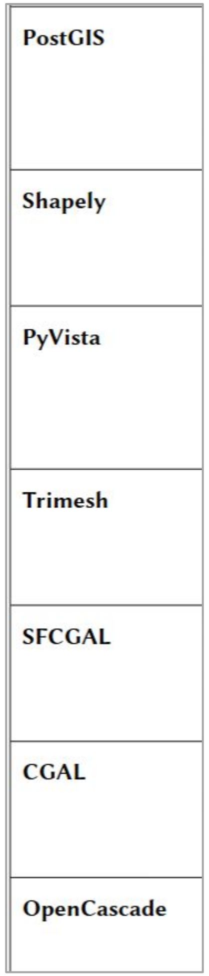
Comparative Analysis of Approaches for Geometric Data Representation in RDF. by Diellza Elshani, Ali Nakhaee, Anthony A. Arrascue, Haris Isakovic, Navid Hedayati, Janakiram Karlapudi, Thomas Wortmann LDAC 2025

Approach III

Integrating RDF-based Systems with Native Geometry Stores or Computer-Graphics Libraries

C) Integrating RDF-based Systems with Native Geometry Stores or Computer-Graphics Libraries



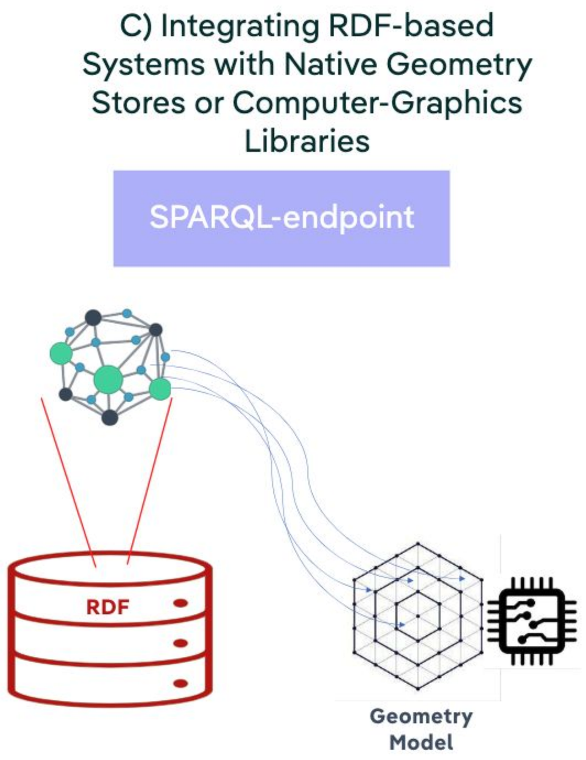


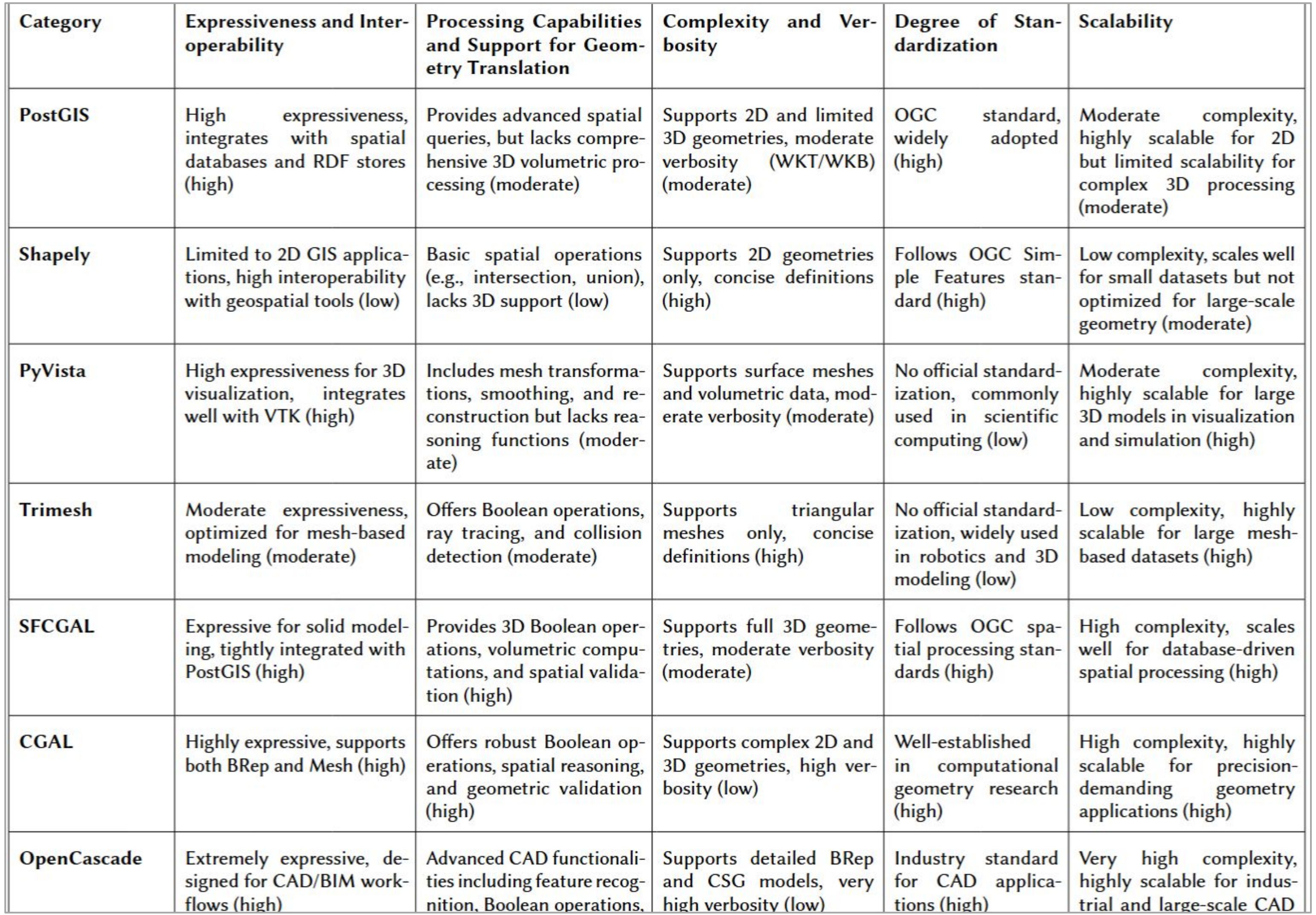
PostGIS
Shapely
PyVista
Trimesh
SFCGAL
CGAL
OpenCascade

Comparative Analysis of Approaches for Geometric Data Representation in RDF. by Diellza Elshani, Ali Nakhaee, Anthony A. Arrascue, Haris Isakovic, Navid Hedayati, Janakiram Karlapudi, Thomas Wortmann LDAC 2025

Approaches to Representing and Querying Geometry in RDF

Integrating RDF-based Systems with Native Geometry Stores or Computer-Graphics Libraries





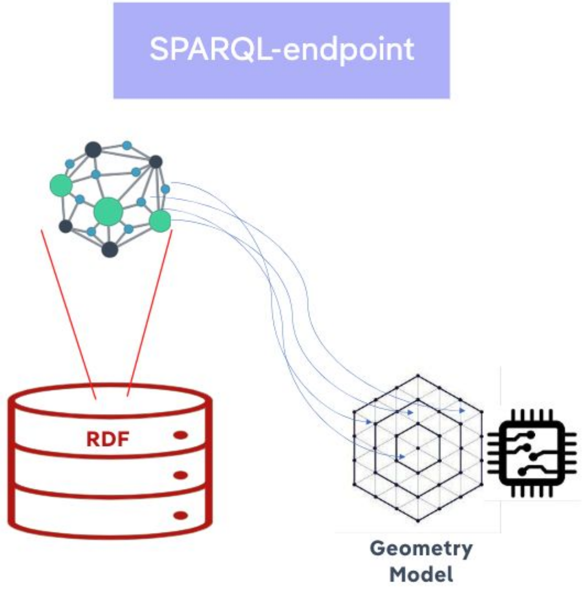
Category	Expressiveness and Interoperability	Processing Capabilities and Support for Geometry Translation	Complexity and Verbosity	Degree of Standardization	Scalability
PostGIS	High expressiveness, integrates with spatial databases and RDF stores (high)	Provides advanced spatial queries, but lacks comprehensive 3D volumetric processing (moderate)	Supports 2D and limited 3D geometries, moderate verbosity (WKT/WKB) (moderate)	OGC standard, widely adopted (high)	Moderate complexity, highly scalable for 2D but limited scalability for complex 3D processing (moderate)
Shapely	Limited to 2D GIS applications, high interoperability with geospatial tools (low)	Basic spatial operations (e.g., intersection, union), lacks 3D support (low)	Supports 2D geometries only, concise definitions (high)	Follows OGC Simple Features standard (high)	Low complexity, scales well for small datasets but not optimized for large-scale geometry (moderate)
PyVista	High expressiveness for 3D visualization, integrates well with VTK (high)	Includes mesh transformations, smoothing, and reconstruction but lacks reasoning functions (moderate)	Supports surface meshes and volumetric data, moderate verbosity (moderate)	No official standardization, commonly used in scientific computing (low)	Moderate complexity, highly scalable for large 3D models in visualization and simulation (high)
Trimesh	Moderate expressiveness, optimized for mesh-based modeling (moderate)	Offers Boolean operations, ray tracing, and collision detection (moderate)	Supports triangular meshes only, concise definitions (high)	No official standardization, widely used in robotics and 3D modeling (low)	Low complexity, highly scalable for large mesh-based datasets (high)
SFCGAL	Expressive for solid modeling, tightly integrated with PostGIS (high)	Provides 3D Boolean operations, volumetric computations, and spatial validation (high)	Supports full 3D geometries, moderate verbosity (moderate)	Follows OGC spatial processing standards (high)	High complexity, scales well for database-driven spatial processing (high)
CGAL	Highly expressive, supports both BRep and Mesh (high)	Offers robust Boolean operations, spatial reasoning, and geometric validation (high)	Supports complex 2D and 3D geometries, high verbosity (low)	Well-established in computational geometry research (high)	High complexity, highly scalable for precision-demanding geometry applications (high)
OpenCascade	Extremely expressive, designed for CAD/BIM workflows (high)	Advanced CAD functionalities including feature recognition, Boolean operations,	Supports detailed BRep and CSG models, very high verbosity (low)	Industry standard for CAD applications (high)	Very high complexity, highly scalable for industrial and large-scale CAD

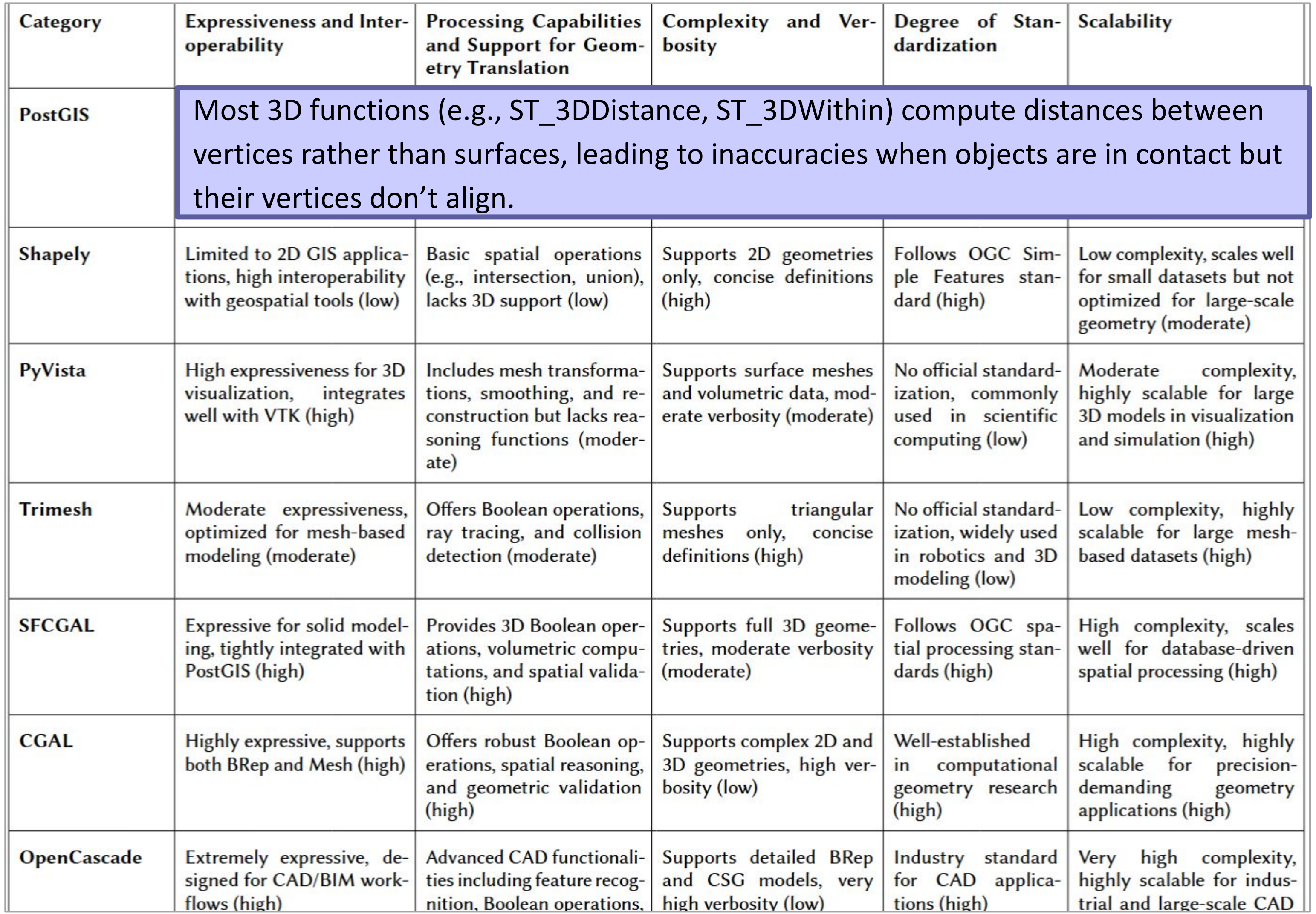
Comparative Analysis of Approaches for Geometric Data Representation in RDF. by Diellza Elshani, Ali Nakhaee, Anthony A. Arrascue, Haris Isakovic, Navid Hedayati, Janakiram Karlapudi, Thomas Wortmann LDAC 2025

Approaches to Representing and Querying Geometry in RDF

Integrating RDF-based Systems with Native Geometry Stores or Computer-Graphics Libraries

C) Integrating RDF-based Systems with Native Geometry Stores or Computer-Graphics Libraries





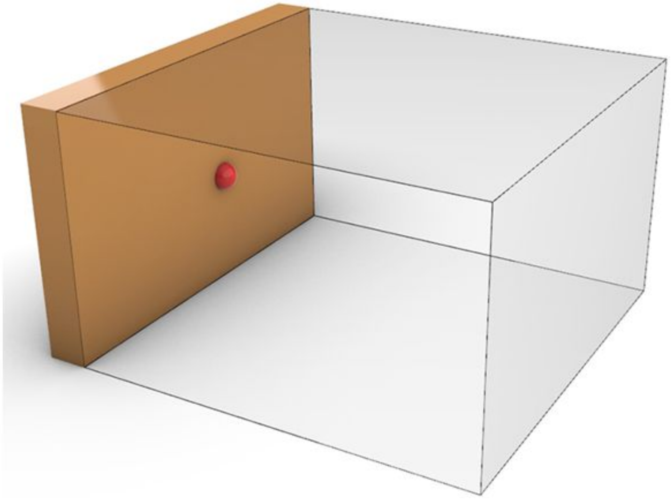
Category	Expressiveness and Interoperability	Processing Capabilities and Support for Geometry Translation	Complexity and Verbosity	Degree of Standardization	Scalability
PostGIS					
Shapely	Limited to 2D GIS applications, high interoperability with geospatial tools (low)	Basic spatial operations (e.g., intersection, union), lacks 3D support (low)	Supports 2D geometries only, concise definitions (high)	Follows OGC Simple Features standard (high)	Low complexity, scales well for small datasets but not optimized for large-scale geometry (moderate)
PyVista	High expressiveness for 3D visualization, integrates well with VTK (high)	Includes mesh transformations, smoothing, and reconstruction but lacks reasoning functions (moderate)	Supports surface meshes and volumetric data, moderate verbosity (moderate)	No official standardization, commonly used in scientific computing (low)	Moderate complexity, highly scalable for large 3D models in visualization and simulation (high)
Trimesh	Moderate expressiveness, optimized for mesh-based modeling (moderate)	Offers Boolean operations, ray tracing, and collision detection (moderate)	Supports triangular meshes only, concise definitions (high)	No official standardization, widely used in robotics and 3D modeling (low)	Low complexity, highly scalable for large mesh-based datasets (high)
SFCGAL	Expressive for solid modeling, tightly integrated with PostGIS (high)	Provides 3D Boolean operations, volumetric computations, and spatial validation (high)	Supports full 3D geometries, moderate verbosity (moderate)	Follows OGC spatial processing standards (high)	High complexity, scales well for database-driven spatial processing (high)
CGAL	Highly expressive, supports both BRep and Mesh (high)	Offers robust Boolean operations, spatial reasoning, and geometric validation (high)	Supports complex 2D and 3D geometries, high verbosity (low)	Well-established in computational geometry research (high)	High complexity, highly scalable for precision-demanding geometry applications (high)
OpenCascade	Extremely expressive, designed for CAD/BIM workflows (high)	Advanced CAD functionalities including feature recognition, Boolean operations,	Supports detailed BRep and CSG models, very high verbosity (low)	Industry standard for CAD applications (high)	Very high complexity, highly scalable for industrial and large-scale CAD

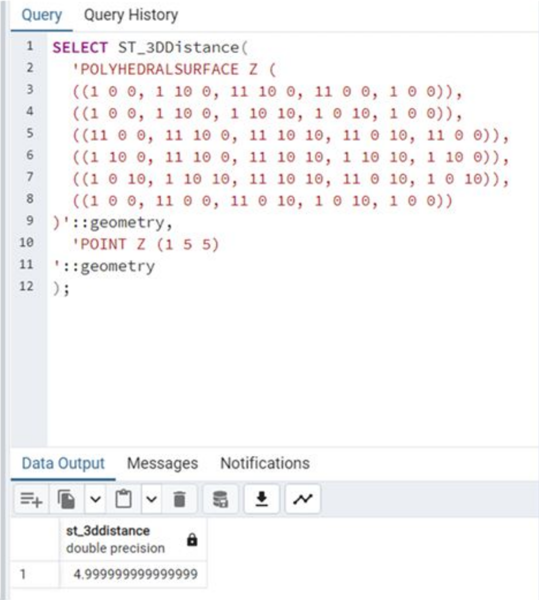
Most 3D functions (e.g., ST_3DDistance, ST_3DWithin) compute distances between vertices rather than surfaces, leading to inaccuracies when objects are in contact but their vertices don't align.

Comparative Analysis of Approaches for Geometric Data Representation in RDF. by Diellza Elshani, Ali Nakhaee, Anthony A. Arrascue, Haris Isakovic, Navid Hedayati, Janakiram Karlapudi, Thomas Wortmann LDAC 2025

Approach III: PostGreSQL, PostGIS

Unexpected 3D distance computation in PostGIS:





```sql
SELECT ST_3DDistance(
  'POLYHEDRALSURFACE Z (
  ((1 0 0, 1 10 0, 11 10 0, 11 0 0, 1 0 0)),
  ((1 0 0, 1 10 0, 1 10 10, 1 0 10, 1 0 0)),
  ((11 0 0, 11 10 0, 11 10 10, 11 0 10, 11 0 0)),
  ((1 10 0, 11 10 0, 11 10 10, 1 10 10, 1 10 0)),
  ((1 0 10, 1 10 10, 11 10 10, 11 0 10, 1 0 10)),
  ((1 0 0, 11 0 0, 11 0 10, 1 0 10, 1 0 0))
)'::geometry,
  'POINT Z (1 5 5)
'::geometry
);
```

	st_3ddistance double precision
1	4.999999999999999

```sql
SELECT ST_3DDistance(
  'POLYHEDRALSURFACE Z (
  ((1 0 0, 1 10 0, 11 10 0, 11 0 0, 1 0 0)),
  ((1 0 0, 1 10 0, 1 10 10, 1 0 10, 1 0 0)),
  ((11 0 0, 11 10 0, 11 10 10, 11 0 10, 11 0 0)),
  ((1 10 0, 11 10 0, 11 10 10, 1 10 10, 1 10 0)),
  ((1 0 10, 1 10 10, 11 10 10, 11 0 10, 1 0 10)),
  ((1 0 0, 11 0 0, 11 0 10, 1 0 10, 1 0 0))
)'::geometry,
  'POLYHEDRALSURFACE Z (
  ((0 0 0, 1 0 0, 1 10 0, 0 10 0, 0 0 0)),
  ((0 0 0, 1 0 0, 1 0 10, 0 0 10, 0 0 0)),
  ((1 0 0, 1 10 0, 1 10 10, 1 0 10, 1 0 0)),
  ((0 10 0, 1 10 0, 1 10 10, 0 10 10, 0 10 0)),
  ((0 0 10, 1 0 10, 1 10 10, 0 10 10, 0 0 10)),
  ((0 0 0, 0 0 10, 0 10 10, 0 10 0, 0 0 0))
)'::geometry
);
```

	st_3ddistance double precision
1	0

PostGIS computes 3D distances using vertices, not full surfaces

Functions affected: `ST_3DDistance`, `ST_3DWithin`, `ST_3DFullyWithin`, `ST_3DIntersects`

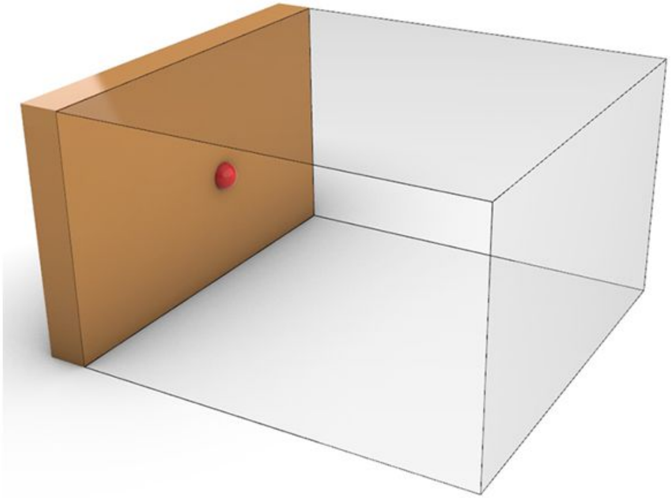
Leads to inaccurate results when nearest points are not vertices

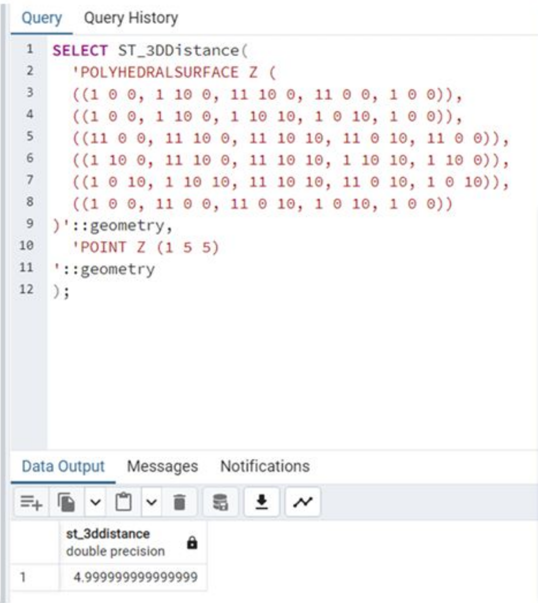
Example: Sphere touching glass wall → expected distance = 0, PostGIS returns ≈ 5

Why? No vertex pair is close enough, despite surface contact

Approach III: PostGreSQL, PostGIS

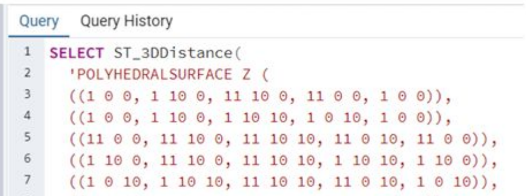
Unexpected 3D distance computation in PostGIS:





```
SELECT ST_3DDistance(
  'POLYHEDRALSURFACE Z (
  ((1 0 0, 1 10 0, 11 10 0, 11 0 0, 1 0 0)),
  ((1 0 0, 1 10 0, 1 10 10, 1 0 10, 1 0 0)),
  ((11 0 0, 11 10 0, 11 10 10, 11 0 10, 11 0 0)),
  ((1 10 0, 11 10 0, 11 10 10, 1 10 10, 1 10 0)),
  ((1 0 10, 1 10 10, 11 10 10, 11 0 10, 1 0 10)),
  ((1 0 0, 11 0 0, 11 0 10, 1 0 10, 1 0 0))
)'::geometry,
  'POINT Z (1 5 5)
'::geometry
);
```

	st_3ddistance double precision
1	4.999999999999999



```
SELECT ST_3DDistance(
  'POLYHEDRALSURFACE Z (
  ((1 0 0, 1 10 0, 11 10 0, 11 0 0, 1 0 0)),
  ((1 0 0, 1 10 0, 1 10 10, 1 0 10, 1 0 0)),
  ((11 0 0, 11 10 0, 11 10 10, 11 0 10, 11 0 0)),
  ((1 10 0, 11 10 0, 11 10 10, 1 10 10, 1 10 0)),
  ((1 0 10, 1 10 10, 11 10 10, 11 0 10, 1 0 10)),
```

maybe it is a modelling question: Solid vs Polyhedral surfaces

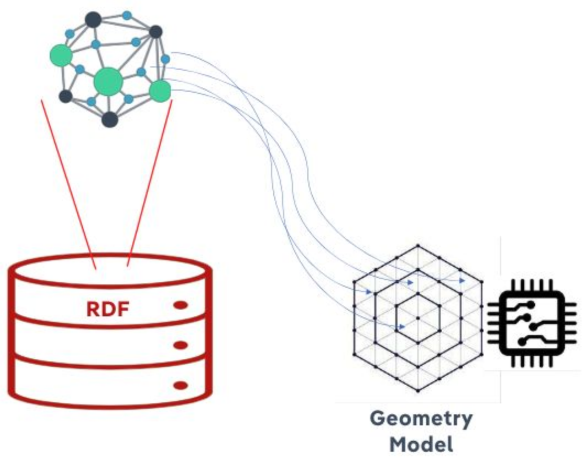
PostGIS computes 3D distances using vertices, not full surfaces
Functions affected: `ST_3DDistance`, `ST_3DWithin`, `ST_3DFullyWithin`, `ST_3DIntersects`
Leads to inaccurate results when nearest points are not vertices
Example: Sphere touching glass wall → expected distance = 0, PostGIS returns ≈ 5
Why? No vertex pair is close enough, despite surface contact

Approaches to Representing and Querying Geometry in RDF

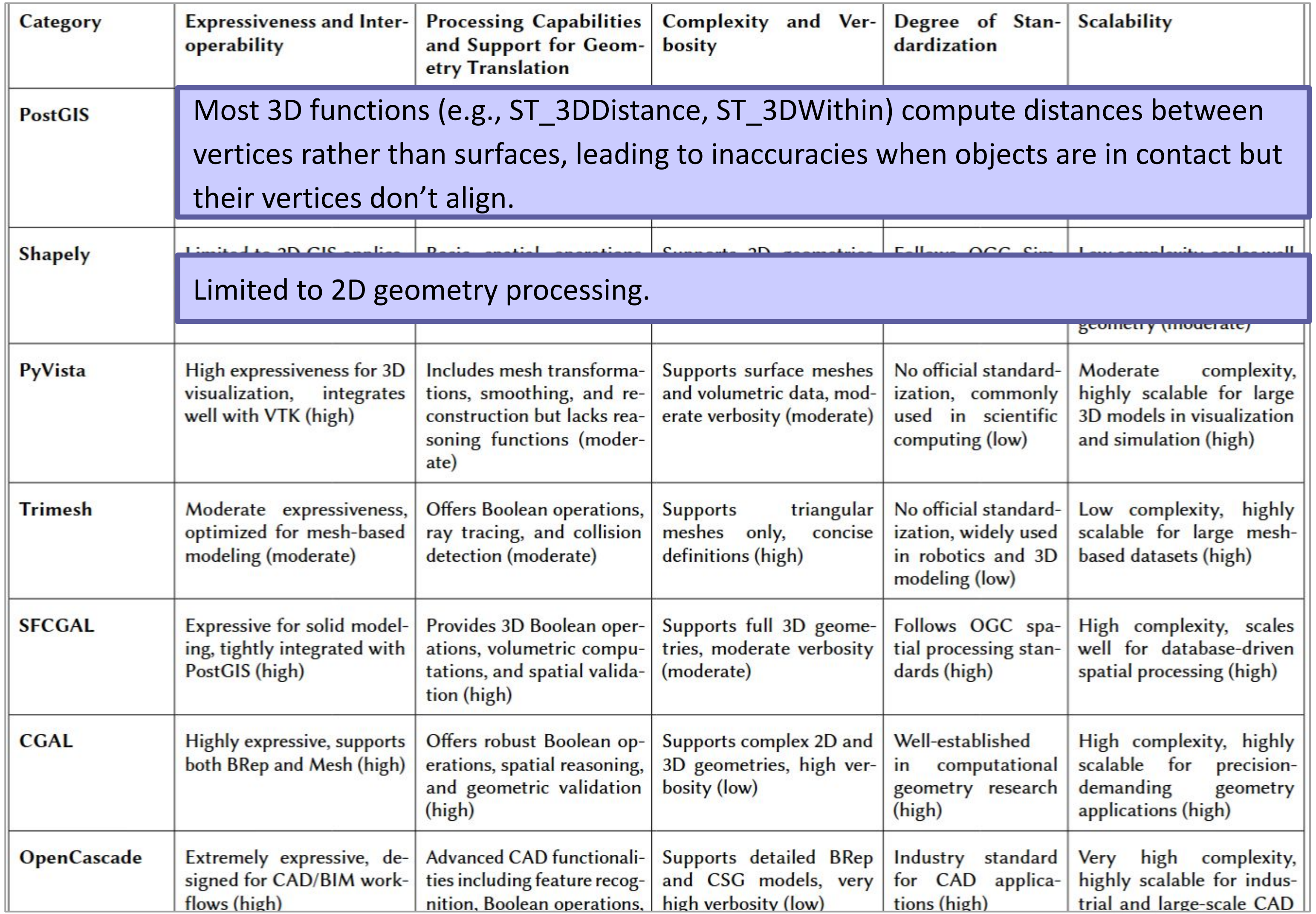
Integrating RDF-based Systems with Native Geometry Stores or Computer-Graphics Libraries

C) Integrating RDF-based Systems with Native Geometry Stores or Computer-Graphics Libraries

SPARQL-endpoint



Comparative Analysis of Approaches for Geometric Data Representation in RDF. by Diellza Elshani, Ali Nakhaee, Anthony A. Arrascue, Haris Isakovic, Navid Hedayati, Janakiram Karlapudi, Thomas Wortmann LDAC 2025



Category	Expressiveness and Interoperability	Processing Capabilities and Support for Geometry Translation	Complexity and Verbosity	Degree of Standardization	Scalability
PostGIS					
Shapely					
PyVista	High expressiveness for 3D visualization, integrates well with VTK (high)	Includes mesh transformations, smoothing, and reconstruction but lacks reasoning functions (moderate)	Supports surface meshes and volumetric data, moderate verbosity (moderate)	No official standardization, commonly used in scientific computing (low)	Moderate complexity, highly scalable for large 3D models in visualization and simulation (high)
Trimesh	Moderate expressiveness, optimized for mesh-based modeling (moderate)	Offers Boolean operations, ray tracing, and collision detection (moderate)	Supports triangular meshes only, concise definitions (high)	No official standardization, widely used in robotics and 3D modeling (low)	Low complexity, highly scalable for large mesh-based datasets (high)
SFCGAL	Expressive for solid modeling, tightly integrated with PostGIS (high)	Provides 3D Boolean operations, volumetric computations, and spatial validation (high)	Supports full 3D geometries, moderate verbosity (moderate)	Follows OGC spatial processing standards (high)	High complexity, scales well for database-driven spatial processing (high)
CGAL	Highly expressive, supports both BRep and Mesh (high)	Offers robust Boolean operations, spatial reasoning, and geometric validation (high)	Supports complex 2D and 3D geometries, high verbosity (low)	Well-established in computational geometry research (high)	High complexity, highly scalable for precision-demanding geometry applications (high)
OpenCascade	Extremely expressive, designed for CAD/BIM workflows (high)	Advanced CAD functionalities including feature recognition, Boolean operations,	Supports detailed BRep and CSG models, very high verbosity (low)	Industry standard for CAD applications (high)	Very high complexity, highly scalable for industrial and large-scale CAD

Most 3D functions (e.g., ST_3DDistance, ST_3DWithin) compute distances between vertices rather than surfaces, leading to inaccuracies when objects are in contact but their vertices don't align.

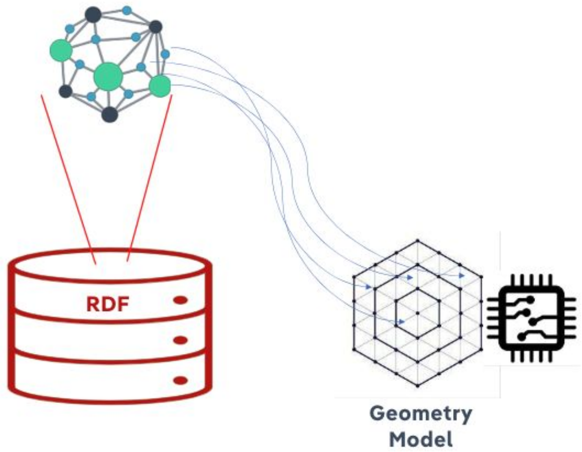
Limited to 2D geometry processing.

Approaches to Representing and Querying Geometry in RDF

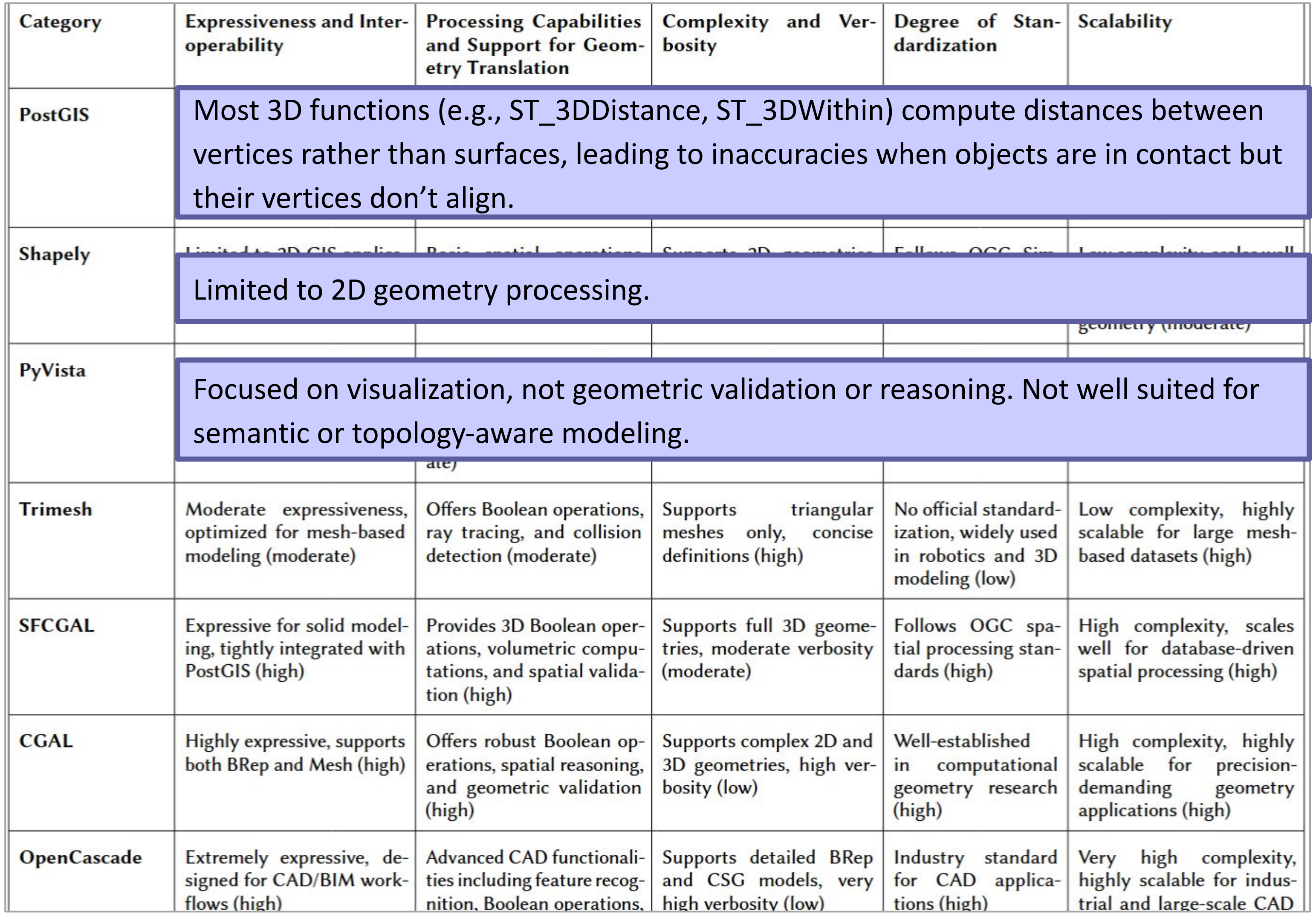
Integrating RDF-based Systems with Native Geometry Stores or Computer-Graphics Libraries

C) Integrating RDF-based Systems with Native Geometry Stores or Computer-Graphics Libraries

SPARQL-endpoint



Comparative Analysis of Approaches for Geometric Data Representation in RDF. by Diellza Elshani, Ali Nakhaee, Anthony A. Arrascue, Haris Isakovic, Navid Hedayati, Janakiram Karlapudi, Thomas Wortmann LDAC 2025



Category	Expressiveness and Interoperability	Processing Capabilities and Support for Geometry Translation	Complexity and Verbosity	Degree of Standardization	Scalability
PostGIS					
Shapely					
PyVista					
Trimesh	Moderate expressiveness, optimized for mesh-based modeling (moderate)	Offers Boolean operations, ray tracing, and collision detection (moderate)	Supports triangular meshes only, concise definitions (high)	No official standardization, widely used in robotics and 3D modeling (low)	Low complexity, highly scalable for large mesh-based datasets (high)
SFCGAL	Expressive for solid modeling, tightly integrated with PostGIS (high)	Provides 3D Boolean operations, volumetric computations, and spatial validation (high)	Supports full 3D geometries, moderate verbosity (moderate)	Follows OGC spatial processing standards (high)	High complexity, scales well for database-driven spatial processing (high)
CGAL	Highly expressive, supports both BRep and Mesh (high)	Offers robust Boolean operations, spatial reasoning, and geometric validation (high)	Supports complex 2D and 3D geometries, high verbosity (low)	Well-established in computational geometry research (high)	High complexity, highly scalable for precision-demanding geometry applications (high)
OpenCascade	Extremely expressive, designed for CAD/BIM workflows (high)	Advanced CAD functionalities including feature recognition, Boolean operations,	Supports detailed BRep and CSG models, very high verbosity (low)	Industry standard for CAD applications (high)	Very high complexity, highly scalable for industrial and large-scale CAD

Most 3D functions (e.g., ST_3DDistance, ST_3DWithin) compute distances between vertices rather than surfaces, leading to inaccuracies when objects are in contact but their vertices don't align.

Limited to 2D geometry processing.

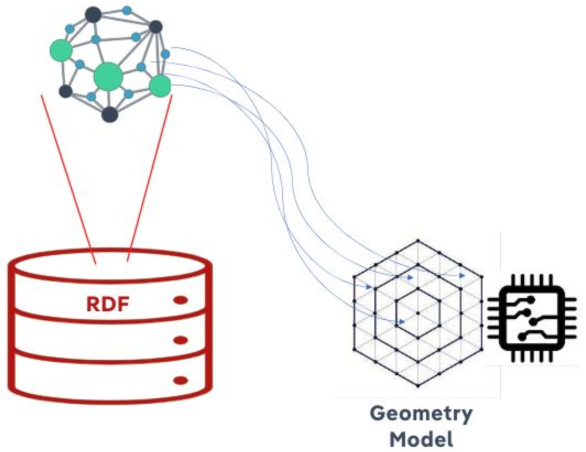
Focused on visualization, not geometric validation or reasoning. Not well suited for semantic or topology-aware modeling.

Approaches to Representing and Querying Geometry in RDF

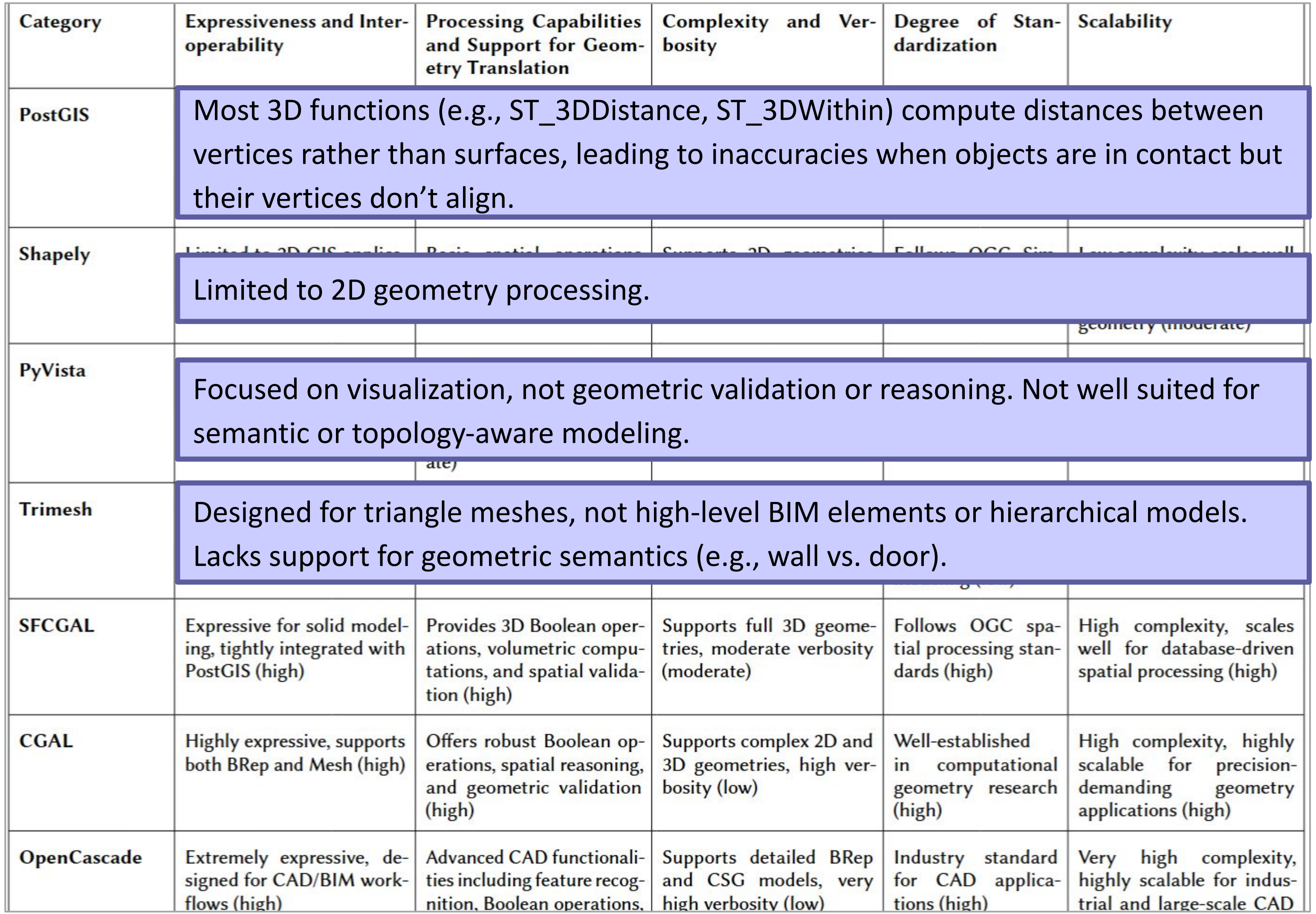
Integrating RDF-based Systems with Native Geometry Stores or Computer-Graphics Libraries

C) Integrating RDF-based Systems with Native Geometry Stores or Computer-Graphics Libraries

SPARQL-endpoint



Comparative Analysis of Approaches for Geometric Data Representation in RDF. by Diellza Elshani, Ali Nakhaee, Anthony A. Arrascue, Haris Isakovic, Navid Hedayati, Janakiram Karlapudi, Thomas Wortmann LDAC 2025



Category	Expressiveness and Interoperability	Processing Capabilities and Support for Geometry Translation	Complexity and Verbosity	Degree of Standardization	Scalability
PostGIS					
Shapely					
PyVista					
Trimesh					
SFCGAL	Expressive for solid modeling, tightly integrated with PostGIS (high)	Provides 3D Boolean operations, volumetric computations, and spatial validation (high)	Supports full 3D geometries, moderate verbosity (moderate)	Follows OGC spatial processing standards (high)	High complexity, scales well for database-driven spatial processing (high)
CGAL	Highly expressive, supports both BRep and Mesh (high)	Offers robust Boolean operations, spatial reasoning, and geometric validation (high)	Supports complex 2D and 3D geometries, high verbosity (low)	Well-established in computational geometry research (high)	High complexity, highly scalable for precision-demanding geometry applications (high)
OpenCascade	Extremely expressive, designed for CAD/BIM workflows (high)	Advanced CAD functionalities including feature recognition, Boolean operations,	Supports detailed BRep and CSG models, very high verbosity (low)	Industry standard for CAD applications (high)	Very high complexity, highly scalable for industrial and large-scale CAD

Most 3D functions (e.g., ST_3DDistance, ST_3DWithin) compute distances between vertices rather than surfaces, leading to inaccuracies when objects are in contact but their vertices don’t align.

Limited to 2D geometry processing.

Focused on visualization, not geometric validation or reasoning. Not well suited for semantic or topology-aware modeling.

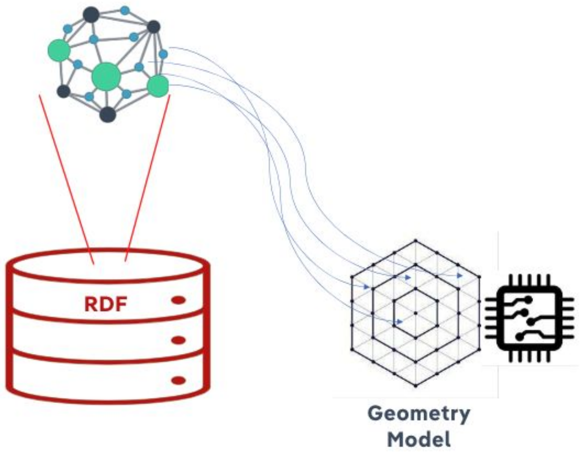
Designed for triangle meshes, not high-level BIM elements or hierarchical models. Lacks support for geometric semantics (e.g., wall vs. door).

Approaches to Representing and Querying Geometry in RDF

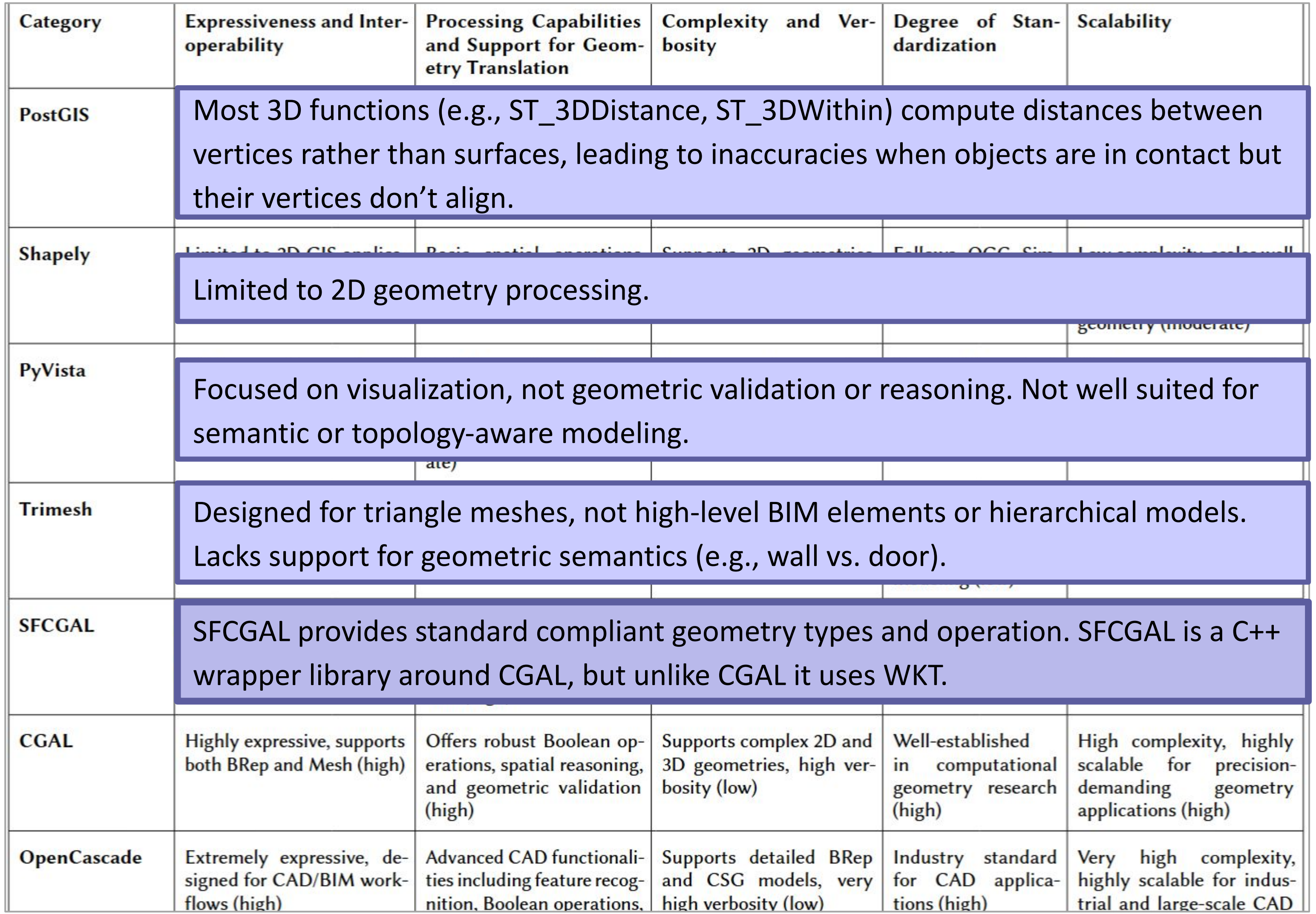
Integrating RDF-based Systems with Native Geometry Stores or Computer-Graphics Libraries

C) Integrating RDF-based Systems with Native Geometry Stores or Computer-Graphics Libraries

SPARQL-endpoint



Comparative Analysis of Approaches for Geometric Data Representation in RDF. by Diellza Elshani, Ali Nakhaee, Anthony A. Arrascue, Haris Isakovic, Navid Hedayati, Janakiram Karlapudi, Thomas Wortmann LDAC 2025



Category	Expressiveness and Interoperability	Processing Capabilities and Support for Geometry Translation	Complexity and Verbosity	Degree of Standardization	Scalability
PostGIS					
Shapely					geometry (moderate)
PyVista		ate)			
Trimesh					
SFCGAL					
CGAL	Highly expressive, supports both BRep and Mesh (high)	Offers robust Boolean operations, spatial reasoning, and geometric validation (high)	Supports complex 2D and 3D geometries, high verbosity (low)	Well-established in computational geometry research (high)	High complexity, highly scalable for precision-demanding geometry applications (high)
OpenCascade	Extremely expressive, designed for CAD/BIM workflows (high)	Advanced CAD functionalities including feature recognition, Boolean operations,	Supports detailed BRep and CSG models, very high verbosity (low)	Industry standard for CAD applications (high)	Very high complexity, highly scalable for industrial and large-scale CAD

PostGIS: Most 3D functions (e.g., ST_3DDistance, ST_3DWithin) compute distances between vertices rather than surfaces, leading to inaccuracies when objects are in contact but their vertices don't align.

Shapely: Limited to 2D geometry processing.

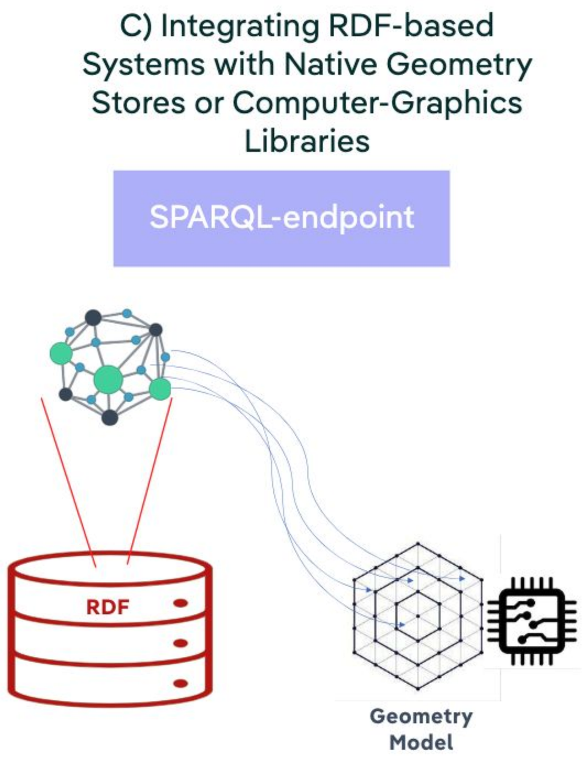
PyVista: Focused on visualization, not geometric validation or reasoning. Not well suited for semantic or topology-aware modeling.

Trimesh: Designed for triangle meshes, not high-level BIM elements or hierarchical models. Lacks support for geometric semantics (e.g., wall vs. door).

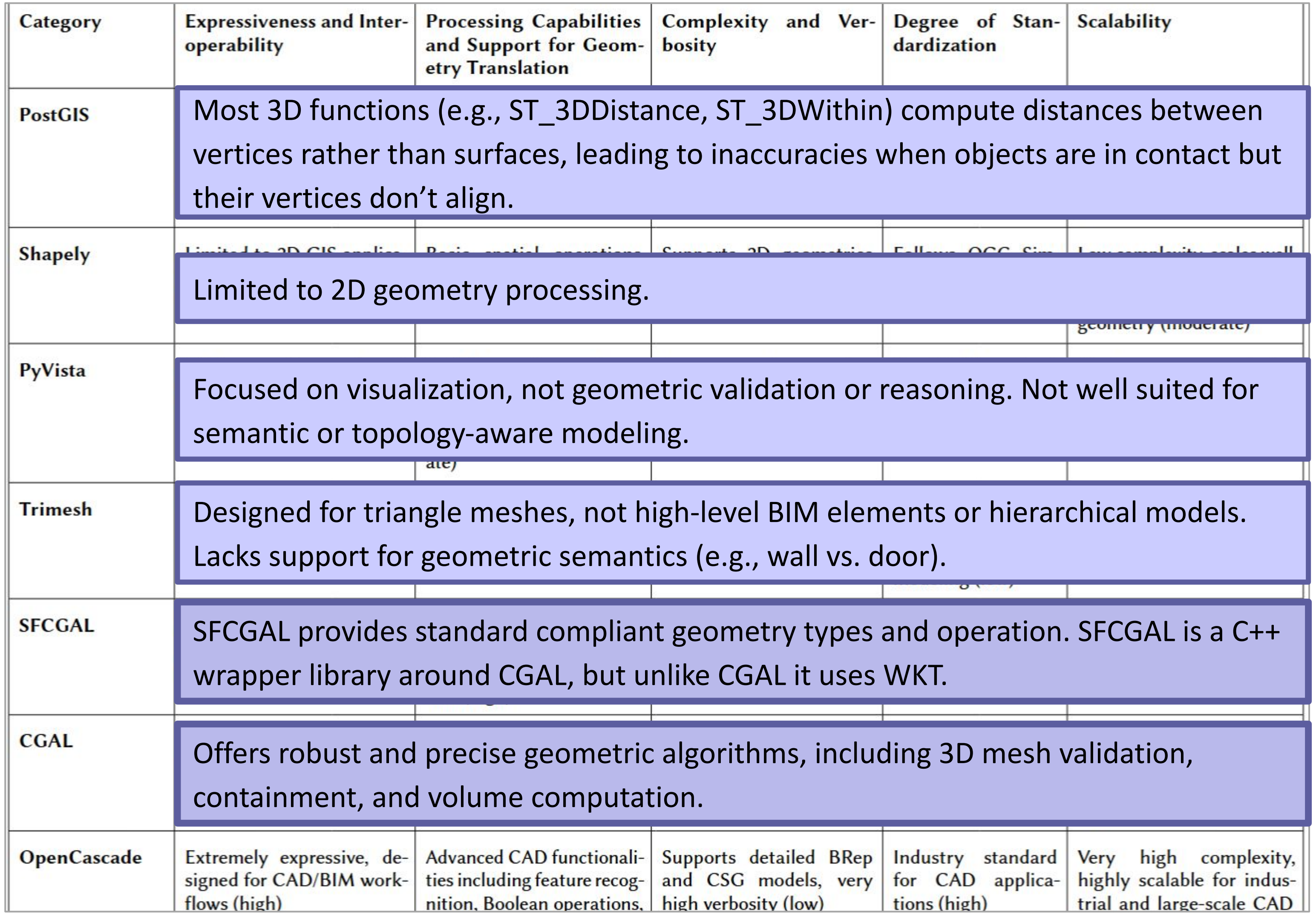
SFCGAL: SFCGAL provides standard compliant geometry types and operation. SFCGAL is a C++ wrapper library around CGAL, but unlike CGAL it uses WKT.

Approaches to Representing and Querying Geometry in RDF

Integrating RDF-based Systems with Native Geometry Stores or Computer-Graphics Libraries



Comparative Analysis of Approaches for Geometric Data Representation in RDF. by Diellza Elshani, Ali Nakhaee, Anthony A. Arrascue, Haris Isakovic, Navid Hedayati, Janakiram Karlapudi, Thomas Wortmann LDAC 2025



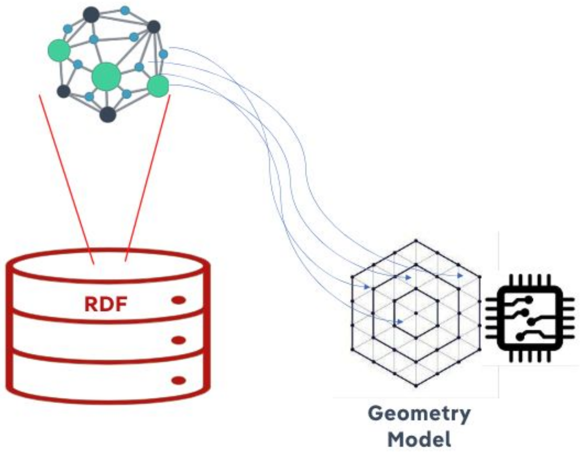
Category	Expressiveness and Interoperability	Processing Capabilities and Support for Geometry Translation	Complexity and Verbosity	Degree of Standardization	Scalability
PostGIS	Most 3D functions (e.g., ST_3DDistance, ST_3DWithin) compute distances between vertices rather than surfaces, leading to inaccuracies when objects are in contact but their vertices don't align.				
Shapely	Limited to 2D geometry processing.				
PyVista	Focused on visualization, not geometric validation or reasoning. Not well suited for semantic or topology-aware modeling.				
Trimesh	Designed for triangle meshes, not high-level BIM elements or hierarchical models. Lacks support for geometric semantics (e.g., wall vs. door).				
SFCGAL	SFCGAL provides standard compliant geometry types and operation. SFCGAL is a C++ wrapper library around CGAL, but unlike CGAL it uses WKT.				
CGAL	Offers robust and precise geometric algorithms, including 3D mesh validation, containment, and volume computation.				
OpenCascade	Extremely expressive, designed for CAD/BIM workflows (high)	Advanced CAD functionalities including feature recognition, Boolean operations,	Supports detailed BRep and CSG models, very high verbosity (low)	Industry standard for CAD applications (high)	Very high complexity, highly scalable for industrial and large-scale CAD

Approaches to Representing and Querying Geometry in RDF

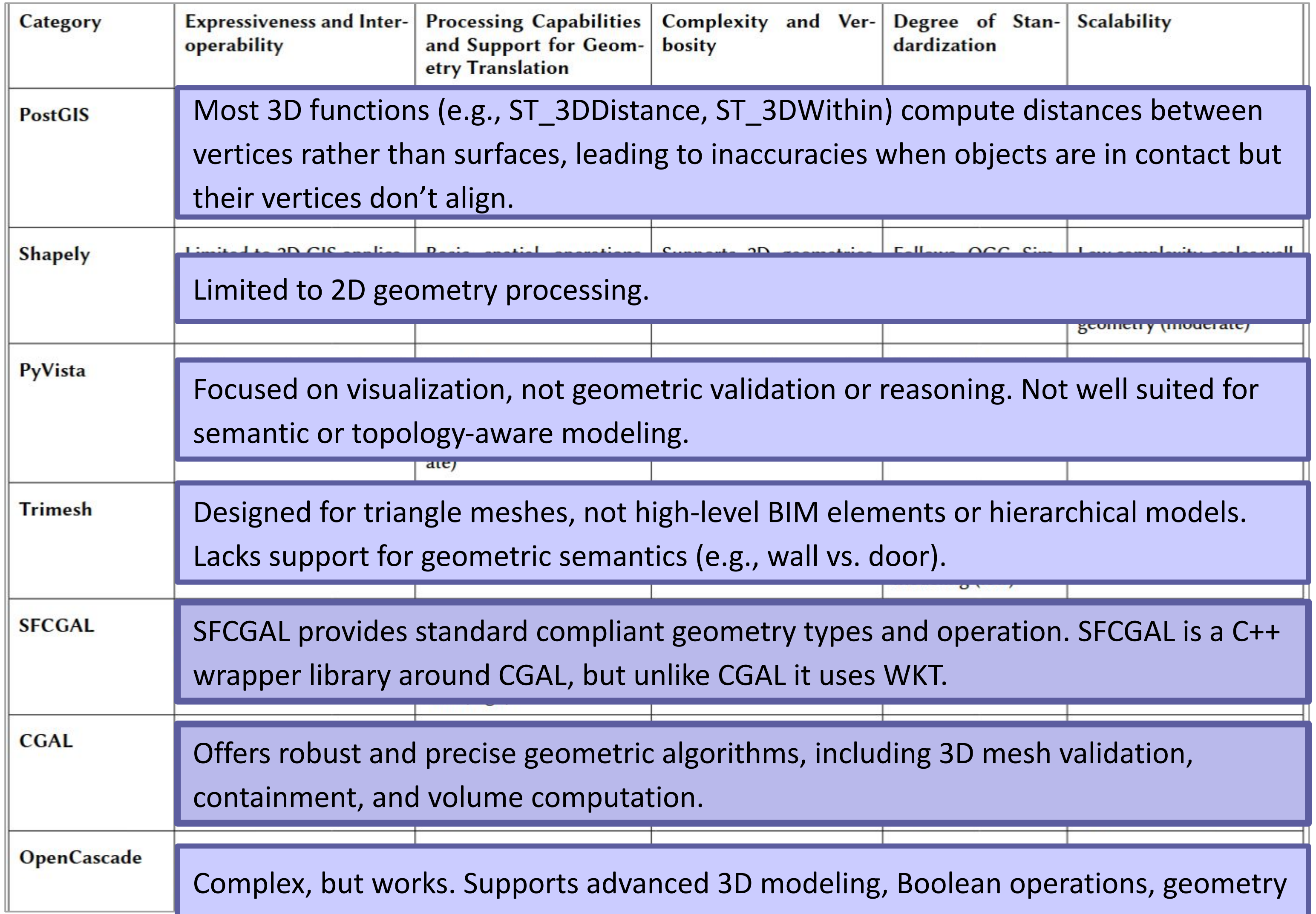
Integrating RDF-based Systems with Native Geometry Stores or Computer-Graphics Libraries

C) Integrating RDF-based Systems with Native Geometry Stores or Computer-Graphics Libraries

SPARQL-endpoint



Comparative Analysis of Approaches for Geometric Data Representation in RDF. by Diellza Elshani, Ali Nakhaee, Anthony A. Arrascue, Haris Isakovic, Navid Hedayati, Janakiram Karlapudi, Thomas Wortmann LDAC 2025



Category	Expressiveness and Interoperability	Processing Capabilities and Support for Geometry Translation	Complexity and Verbosity	Degree of Standardization	Scalability
PostGIS					
Shapely					
PyVista					
Trimesh					
SFCGAL					
CGAL					
OpenCascade					

Most 3D functions (e.g., ST_3DDistance, ST_3DWithin) compute distances between vertices rather than surfaces, leading to inaccuracies when objects are in contact but their vertices don’t align.

Limited to 2D geometry processing.

Focused on visualization, not geometric validation or reasoning. Not well suited for semantic or topology-aware modeling.

Designed for triangle meshes, not high-level BIM elements or hierarchical models. Lacks support for geometric semantics (e.g., wall vs. door).

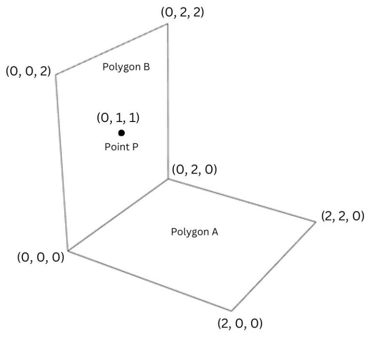
SFCGAL provides standard compliant geometry types and operation. SFCGAL is a C++ wrapper library around CGAL, but unlike CGAL it uses WKT.

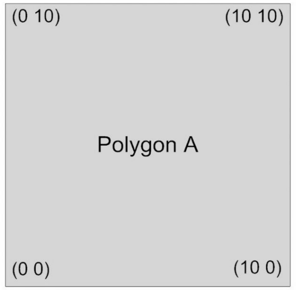
Offers robust and precise geometric algorithms, including 3D mesh validation, containment, and volume computation.

Complex, but works. Supports advanced 3D modeling, Boolean operations, geometry validation, and solid construction; ideal for CAD-grade workflows.

Detailed Evaluation of Approaches

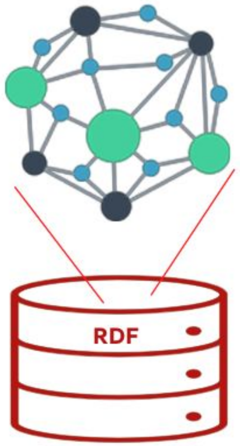
Benchmark	**RDF-based Geometry Descriptions** Eg: ifcOWL	**Non-RDF Geometry as RDF Literals** Eg: GeoSPARQL	**Linking to Non-RDF Geometry Files** Eg: PostGIS
Expressiveness	Highly expressive for BIM data with rich semantics but verbose for geometry.	Supports rich spatial vocabulary for 2D geometries and relationships.	Extensive geometric data types with additional attributes for geographic features.
Complexity	Complex due to IFC-to-OWL translation and reasoning over large datasets.	Easy for basic queries; advanced spatial reasoning can be challenging.	Simple for basic tasks; advanced functionalities require deeper knowledge.
Geometry Support	Mirrors IFC schema but verbose; limited efficiency for complex geometries.	Supports 2D geometries (points, lines, polygons); lacks 3D support.	Robust 2D geometry support with some 3D functionality.
Geometrical Functions	No native geometrical functions.	Comprehensive 2D functions (buffers, intersections, distances); lacks 3D functions.	Extensive 2D functions; growing 3D capabilities but still limited.
Geometry Validation	No inherent validation capabilities.	No built-in validation; some implementations integrate external tools for 2D.	Built-in 2D validation; limited support for 3D integrity checks.
Scalability	Limited scalability due to RDF verbosity and computational overhead.	Scales well with optimized RDF stores; performance varies with implementation.	Highly scalable with spatial indexing and PostgreSQL integration.
Interoperability	Strong semantic interoperability; challenges with consistent geometric representation.	Interoperable via OGC standards for data exchange.	Adheres to OGC standards, enabling broad system integration.
Standardization	Benefits from IFC and OWL standards; requires ongoing standardization efforts.	Well-established by OGC with active development.	Strong alignment with OGC standards; stable implementation.

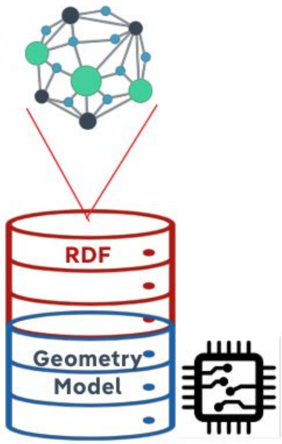


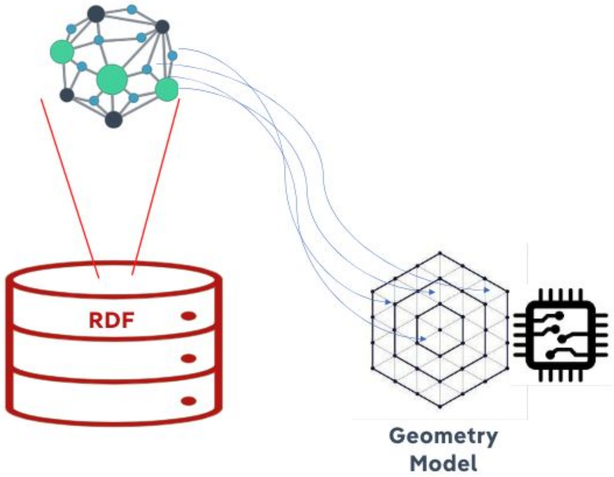


Comparative Analysis of Approaches for Geometric Data Representation in RDF. by Diellza Elshani, Ali Nakhaee, Anthony A. Arrascue, Haris Isakovic, Navid Hedayati, Janakiram Karlapudi, Thomas Wortmann LDAC 2025

Evaluation of Approaches







RDF-based Geometry Modeling: Represents geometry directly in RDF using vocabularies like ifcOWL or GeoSPARQL; highly expressive but lacks processing and validation capabilities.

Extending Triplestore and Querying Capabilities: Adds spatial functions to RDF triplestores (e.g., GeoSPARQL) to enable geometry querying; supports some in-graph computations, some limited to 2D. But they are an efficient method to query.

Integrating RDF-based Systems with Native Geometry Stores or Graphics Libraries: Links RDF to external systems like PostGIS or CGAL for advanced geometry processing; offers strong 3D support but requires integration, propagating data, where data might be disconnected / challenges with depended parameters.

Comparative Analysis of Approaches for Geometric Data Representation in RDF. by Diellza Elshani, Ali Nakhaee, Anthony A. Arrascue, Haris Isakovic, Navid Hedayati, Janakiram Karlapudi, Thomas Wortmann LDAC 2025

Summary: Bridging Geometry Processing and RDF

- Dedicated query languages like GeoSPARQL or BimSPARQL enable efficient, domain-specific geometry processing and semantic inference within RDF systems.

- **External geometry libraries** like CGAL and OpenCascade offer good support for 3D operations (They enable **precise validation**, **Constructive Solid Geometry (CSG)**, and **spatial analysis** beyond RDF-native capabilities). However, they remain **separate from RDF**, requiring external integration.

- **SPARQL can be extended** with custom functions to invoke geometry processing as needed. This creates a **scalable and flexible pathway** to support advanced geometry in RDF-based systems.

Thank you!

Diellza Elshani
diellza.elshani@icd.uni-stuttgart.de

Research Associate | Ph.D. Candidate
Chair for Computing in Architecture
Institute for Computational Design and Construction (ICD/CA)
Faculty of Architecture and Urban Planning
Universität Stuttgart
Keplerstr. 11 | 70174 Stuttgart | Germany