A system for querying spatial relations between technical drawings and geometric data using SPARQL

Fabian Rolke[1]

[1]Chair of Design Computation (DC), RWTH Aachen University, Aachen, Germany

E-mail(s): rolke@dc.rwth-aachen.de

Abstract: The increasing size and heterogeneity of construction related project databases demands significant effort from a limited number of data experts in the architecture, engineering, construction, and operation (AECO) industry. Advancements in technologies for capturing building data necessitate ongoing development in transformation, integration, and interoperability. A key objective is linking heterogeneous building data through their respective spatial alignment. While several RDF and SPARQL extensions support two-dimensional relational querying, there is a notable lack of technologies capable of processing three-dimensional data. Technical drawings are still one of the most common mediums for transferring geometrical data between stakeholders due to their abstract, reduced representational quality, high information density, and extensive use on construction sites. However, they often exist in non-machine-readable formats that hinder queries on depicted data such as location, rotation, scale relative to built assets, and individual annotations. This paper presents an approach to link and query topological, directional, and metric relationships among 2D plans and other data types by storing their three-dimensional spatial information as RDF. A spatial calculus identifies correlations between selected entities via SPARQL queries. Results are written back in RDF to enrich existing databases or provide specific answers about entity relationships.

Keywords: RDF, SPARQL, technical drawing, spatial querying, geometry processing

DOI: 10.18154/RWTH-CONV-254922. Published in the conference proceedings of the 36. Forum Bauinformatik 2025, Aachen, Germany,

1 Introduction

One of the biggest tasks in the architecture, engineering, construction, and operation industry is addressing the challenges of closing the gap between the real and digital world. For this purpose, conceptualizing spatial relations between entities is just as significant as defining their geometrical representations. All spatial relations, whether real or digital, are classified as either topological, directional, or metric [1] [2]. Randell et al. provide an overview of the base topological relations: disjoint, meet, overlap, equal, coveredBy, inside, covers, and contains [3]. Building on this work, the Relative Location (RELOC) Ontology by Göbels and Beetz allows for the conversion of natural-language-based location descriptions into RDF-structured spatial relationships, thus making them machine-readable. Topological relations such as meet, intersect, and containedIn are combined with directional relations such as front, right, or vertical center, allowing for spatial specification in RDF without the immediate need for vector data serializations [4]. Göbels and Schulz further outlined the need for a spatial vocabulary enabling classification of different space types when working with large, heterogeneous datasets [5]. An Entity Space, defined as a singular, spatial "thing", is a subclass of Asset Space which is any kind of overall reference model of a building, or larger building part (e.g., IFC model, PCD). Entity Space has the subclasses Point Space, Area Space (vertical or horizontal) and Volume Space [5]. While these concepts and classifications apply to almost all spatial elements, one in particular holds my special personal interest. "Technical drawings contain information about the complete geometry of the building. In contrast to point clouds, plans also show the dimensions of non-visible components, such as the foundation. Plans are therefore indispensable for extracting the entire building geometry." [6, p. 31]. Their ability to hold extensive geometric and topological data through a set of lines and its widespread use on-site [7] makes them arguably the primary tool for spatial information transfer in today's AECO industry. A spatial relation processor could potentially allow projections of different data types onto technical drawings, prompting the research question:

"How can spatial relations between technical drawings and geometric data be identified through spatial alignment to enable semantic data transfer?"

To subdivide the objective, the research question can be broken down into these sub-questions:

"As what does the virtual geometric representation of a technical drawing classify?"

"How should geometric data be stored in RDF?"

"How can spatial relations be calculated based on RDF data as input?"

To resolve the research questions, the design science research method was employed. By applying its iterative cycles of analysis, development, evaluation and diffusion, a framework was conceptualized, comprising several smaller proof-of-concept-functions for data transformations and geometric operations. These analysis, evaluation and diffusion cycles helped to outline competencies and limitations of the developed system, continually shaping and shifting its functionality to address the research question.

2 Related Work

Çelik et al. conducted a comparative analysis of a hand-defined filter-based (HDFs) and a CycleGAN-based cleanup method for technical drawings. Their findings indicate that the CycleGAN model outperforms HDFs in redrawing lines, geometric shapes, text units, and numbers, thereby enhancing edge detection (e.g., with Sobel filters) and improving OCR accuracy. [8]. Recent works by Faltin et al. present a retrospective modelling approach based on spatially arranged drawing views. Utilizing a Keypoint R-CNN (Region-based Convolutional Neural Network), they were able to simultaneously detect elevation and section symbols. Subsequently, they were able to automatically determine the 2D plans spatial alignment relative to each other [9] [10]. Their work paves the way for projecting geometric data onto technical drawings thorugh respective spatial alignment. The remaining questions on how to store a plans spatial data in a machine readable format and how that data can be queried or linked to other entities is adressed by Zhang and Beetz: BimSPARQL is a framework extending

SPARQL's querying capabilities in regard to building data [11]. "Unlike GeoSPARQL, it is an academic research proposal on incorporating building-related semantics. It introduced domain-specific functions for schema-level semantics, instance-level semantics, and spatial reasoning" [12, p. 8]. Similar to BimSPARQL, Nikitopoulos et al. present the "first parallel and scalable in-memory solution to the problem of spatio-temporal SPARQL query processing" [13, p. 648]. Their proposed DiStRDF System (Distributed spatiotemporal RDF engine) comprises a processing and storage layer. Standard SPARQL queries and separate spatio-temporal constraints are computed by a processing layer which communicates with a storage layer, responsible for distributed storage of RDF data [13]. A Spatial query system utilizing a multi-LLM agent framework, allowing natural language relation and entity queries through triangulated bounding boxes of IFC elements, has been presented by Li et al. [14]. Limited to concentrated geometric data of structural elements (e.g., wall, beam) and general spatial queries, this paper still recommends research in the field of automatic inferencing of missing spatial information in BIM models. Their work furthers the idea of verbally specifying spatial data and queries.

3 Methodology

To analyse and evaluate spatial relations between various building project data types, a three-dimensional virtual test environment was established. Technical drawings were spatially aligned using point cloud data as a topological reference, employing basic transformation commands: location, rotation, and scaling. The 3D geometry of building components depicted in technical drawings was created, resembling the geometric foundation of IFC-elements. Fictional damages were modeled as bounding boxes, representing the result of a conversion process developed by Göbels and Beetz that translates natural language damage reports into geometries [15]. Figure 1 shows the mentioned data-types, superimposed and projected onto a technical drawing. This model was based on a dataset of the Aachen Cathedral, Germany.

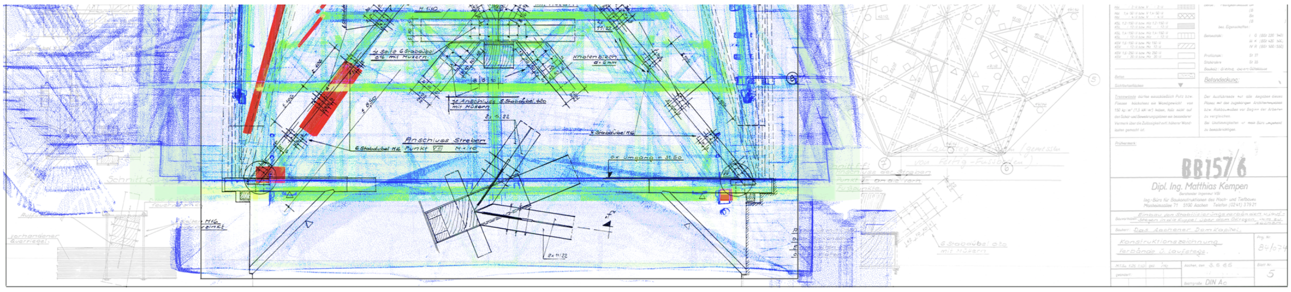

Figure 1: Heterogeneous building data superimposed and projected onto technical drawing. Blue=PCD, green=IFC-elements, red=Damage OBBs.

3.1 Data quality

Despite *limited data heterogeneity* necessitating the creation of additional artificial geometric content to examine potential topological relations between technical drawings and other types of data, the Aachen Cathedral dataset was well-suited for this task due to several factors. It contains a *large number of high-resolution technical drawings* with varying scales, viewing directions, and annotations such as text, measurements, and axes. *Extensive point cloud data* (PCD) provides complete coverage of both the roof's interior and exterior, serving as an accurate spatial reference (asset) for all entities.

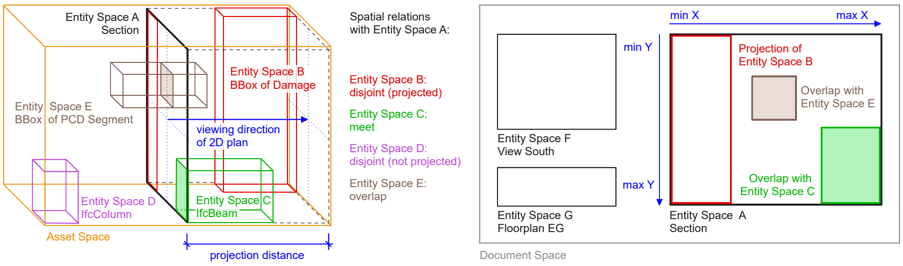

Figure 2: Depiction of technical drawing inside asset and document with spatial relation to multiple entities as well as its projection distance.

The *structural complexity* enables a comprehensive analysis of edge cases in technical drawing orientation and informs a broadly applicable alignment method.

3.2 Accuracy of spatial alignment

The experimental approach highlighted the practicality of spatial alignment of entities with component-specific accuracy to minimize data quality loss during information transfer onto technical drawings. A study on dynamic accuracy levels based on drawing scales and desired information clarity exceeds the scope of this work; therefore, a recommendation for future research is proposed. For now, metric measurements in this study are presented as approximately x.xx m.

3.3 Topological relations

While spatial relations of technical drawings and other data types can be expressed with the vocabulary, presented by Randell et al. [3], only a subset shows relevance when examining the test environment. Technical drawings represent planes in 3D space. Relations such as equal, covers, inside, and contains are rare and typically arise when multiple drawings occupy the same location and share axis alignment. For instance, a vertical section may cover a construction detail of a specific sub-area (coveredBy). Alternatively, two plans may be geometrically and topologically equal by sharing location, axis alignment, and scale while differing in attribute data. The most frequently observed relations are disjoint, meet, and overlap. The relation coveredBy mostly classifies the connection between a technical drawing encapsulated by a bounding box representing a building space. Thus, functions emerging from this work for determining spatial relations should primarily focus on classifying disjoint, meet, and overlap.

3.4 Geometric characteristics of technical drawings

The specification of technical drawings as vertical or horizontal Area Spaces is geometrically correct but topologically insufficient. Some technical drawings do not overlap with all depicted entities, making the calculation of linked elements difficult. A clarification on their spatial extent is needed. Their geometric representation, a plane, in addition to its max projection distance creates an aligned bounding box which encapsulates the plan's spatial influence, rendering the geometric understanding of technical drawings for this work as an Area Space parented to a Volume Space, or just a Volume Space (Fig. 2).

3.5 Conceptualizing a spatial query and processing framework

The following competency questions (CQ) frame the desired functional extent of the spatial query system for a heterogeneous spatial building dataset. CQ1: "Select all bounding boxes of damages classified as water damages which overlap with the Volume Space defining the spatial influence of Drawing A". CQ2: "Select technical drawings with a local z to global -z axis alignment, overlapping with Entity Spaces attributed with maintenance work". A system capable of adressing these questions was framed (Fig. 3).

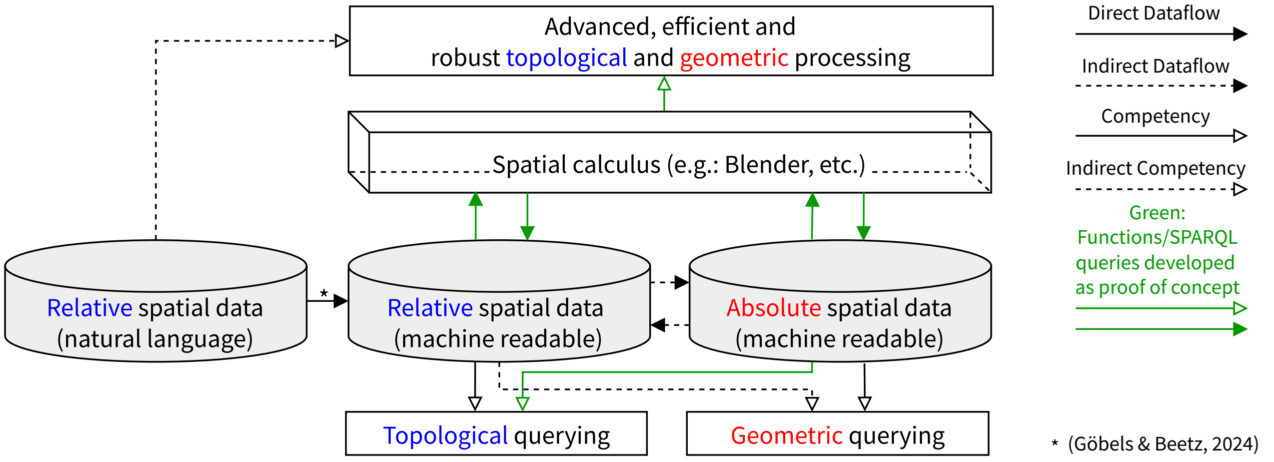

Figure 3: Spatial query and processing Framework

4 Results

The following table 1 shows the results of a comparative analysis of overlap detection algorithms in different environments: (1) Blender python script, (2) SPARQL query.

Table 1: Processing times of overlap detection in milliseconds in a one against all scenario. No.E.S=Number of Entity Spaces, BL=Blender, SP=SPARQL, CE=CenterExtent-BoundingBox algorithm, SP=Sweep and Prune algorithm, MM=MinMax algorithm

No.E.S.	BL.CE.	BL.SP.	SP.CE.	SP.MM.
10	0	0	160	140
55	10	5	209	154
105	19	10	278	174
303	24	30	474	308
507	40	40	645	419

5 Discussion

As what does the virtual geometric representation of a technical drawing classify? For the objective of calculating spatial relations based on the maximum spatial extent, the virtual geometric representation of a technical drawing should be classified as a bounding box or rectangular volumetric space, as specified in section 3.4.

How should geometric data be stored in RDF? While Haverkort [16] outlines the advantage of using simplified representations (bounding boxes) of geometric data, Bonduel et al. [17] demonstrate how linking of multiple geometric representations allows geometric processing on different levels of detail. These approaches sparked the evaluation of different bounding box types regarding the efficiency of detecting overlaps between them using the presented framework. In the Blender python scripts, as well as the SPARQL queries, the MinMax bounding box (AABB) performed best, as the maximum and minimum values, needed for detecting overlaps are already precomputed. By incorporating axis alignment statements, Entity Spaces can be aligned relative to other entities or Asset Spaces. This bounding box RDF-structure matches with the current working draft by Göbels et al. [5].

How can spatial relations be calculated based on RDF data as input? The comparative analysis' results outlined SPARQL's poor efficiency and accuracy for querying overlaps between Entity Spaces, especially in all against all scenarios. The purposefulness of importing simplified geometric data for complex spatial processing was emphasized. A first rough selection, made by a SPARQL query could filter relevant entities for import. A second more accurate pass inside the spatial calculus could sort out false positives. Their geometric representations could then be used for more complex geometrical or topological processing: Enriching triple-stores with spatial links, transferring semantics based on spatial filters or projecting entities' geometric data onto other Entity- or Document Spaces e.g., technical drawings.

Algorithm 1: Exemplary spatial data export as RDF in compliance with DIN EN ISO 21597-2

```
ex:Section_A-A a ex:AreaSpace ;
        icdd:hasLinkElement ex:Section_A-A_LinkElement .
ex:EntitySpace_1 a ex:VolumeSpace ;
        ex:componentType ex:Beam .
        icdd:hasLinkElement ex:EntitySpace_1_LinkElement .
ex:SpatialLink a icdd:DirectedLink ;
        icdd:hasFromLinkElement ex:EntitySpace_1_LinkElement ;
        icdd:hasToLinkElement ex:Section_A-A_LinkElement .
```

Algorithm 2: SPARQL query retrieving the component type of an Entity Space, depicted on a technical drawing via an ICDD-link

```
SELECT ?plan ?entitySpace ?projectedComponentType
WHERE {
    ?link icdd:hasToLinkElement ?toLinkElement ;
          icdd:hasFromLinkElement ?fromLinkElement .
    ?plan icdd:hasLinkElement ?toLinkElement ;
          rdf:type ex:AreaSpace .
    ?entitySpace icdd:hasLinkElement ?fromLinkElement ;
                 ex:componentType ?projectedComponentType .
}
```

Based on the comparative analysis results and the analysis of spatial structures and characteristics of technical drawings in R^3, a framework of export, import, and processing functions is introduced, transforming data from spatial vocabularies into geometries for advanced processing (insert figure). Similar to DiStRDF [13], the system separates storage and processing. Unlike Apache Spark [18], it imports spatial RDF data into a spatial calculus, supporting diverse geometric and topological operations. Blender was used as an exemplary calculus via its Python API. In theory, the developed functions could be enhanced to handle more complex spatial relation processing and integrated into a single, software-agnostic function, improving portability. Besides the vendor dependency of the current framework draft, limiting its diffusion or implementation as a web service, the system does not work with oriented bounding boxes, which would enable more accurate geometric processing.

6 Conclusion

This paper presents a system for bridging gaps between relative and absolute location data in hopes of advancing the broader objective of linking semantic data and projecting geometric data through respective spatial alignment. The framework consists of an RDF-to-geometry converter, a spatial relation processor, and a geometry-to-RDF converter (Fig. 3).

Future work regarding comparative analysis of spatial indexing methods could enable the implementation of spatial functions via standard-compliant SPARQL queries without needing an external processing engine. For an advanced diffusion of the system, its implementation as spatial functions, comprising a web service, could allow clients to communicate with a database and a spatial processing engine that returns query results or updates the RDF database based on entity queries and relational queries as input.

Converting an Entity Space's three-dimensional Cartesian coordinates to local technical drawing pixel coordinates or normalized values (for vector-based formats) would allow projecting geometric data onto technical drawings. Developing a viewer to overlay an Entity Space's local pixel coordinates on the corresponding technical drawing could enhance usability and promote sustainable dissemination within the industry.

Acknowledgement This research is part of the Raumlink project funded by the German Research Foundation (DFG) – Project number 501812634.

References

[1] D. Papadias and Y. Theodoridis, "Spatial relations, minimum bounding rectangles, and spatial data structures", *International Journal of Geographical Information Science*, vol. 11, Mar. 1997. DOI: 10.1080/136588197242428

[2] R. Chbeir, Y. Amghar, and A. Flory, "Novel indexing method of relations between salient objects", in *Effective Databases for Text & Document Management*, IGI Global, 2003.

[3] B. Nebel, C. Rich, and W. R. Swartout, *Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference (KR '92)*. M. Kaufmann, 1992.

[4] A. Göbels and J. Beetz, "Relative Location Ontology: An ontological model for representing directional topological relationships between spatial entities in oriented space", in *LDAC 2024 – 12th Linked Data in Architecture and Construction Workshop*, Bochum, Germany, 2024.

[5] A. Göbels, O. Schulz, and J. Beetz, "Towards a Common Digital Space: Proposing a Schema for Spatially Linking Heterogeneous Resources", in *Proceedings of the 41st International Conference of CIB W78*, Morocca, Marrakech, 2024.

[6] A. Borrmann et al., "Technologien zur Generierung digitaler Zwillinge als Grundlage für Betrieb und Instandhaltung baulicher Infrastruktur", 2022.

[7] K. A. Bisharat, *Construction Graphics: A Practical Guide to Interpreting Working Drawings*. John Wiley & Sons, Mar. 2025.

[8] F. Çelik, B. Faltin, and M. König, “Improving 2D Construction Plans with Cycle-Consistent Generative Adversarial Networks”, May 2022. DOI: 10.1061/9780784483893.007

[9] B. Faltin, P. Schönfelder, and M. König, “Inferring Interconnections of Construction Drawings for Bridges Using Deep Learning-based Methods”, in *ECPPM 2022 - eWork and eBusiness in Architecture, Engineering and Construction 2022*, CRC Press, 2023.

[10] B. Faltin, P. Schönfelder, D. Gann, and M. König, “Reconstructing as-built beam bridge geometry from construction drawings using deep learning-based symbol pose estimation”, *Advanced Engineering Informatics*, vol. 62, Oct. 2024. DOI: 10.1016/j.aei.2024.102808

[11] C. Zhang, J. Beetz, and B. de Vries, “BimSPARQL: Domain-specific functional SPARQL extensions for querying RDF building data”, *Semantic Web*, vol. 9, Sep. 2018. DOI: 10.3233/SW-180297

[12] D. Elshani et al., “Comparative Analysis of Approaches for Geometric Data Representation in RDF”, 2025.

[13] P. Nikitopoulos, A. Vlachou, C. Doulkeridis, and G. A. Vouros, “Parallel and scalable processing of spatio-temporal RDF queries using Spark”, *GeoInformatica*, vol. 25, Oct. 2021. DOI: 10.1007/s10707-019-00371-0

[14] A. Li, P. K.-Y. Wong, X. Tao, J. Ma, and J. C. P. Cheng, “An interactive system for 3D spatial relationship query by integrating tree-based element indexing and LLM-based agent”, *Advanced Engineering Informatics*, vol. 66, Jul. 2025. DOI: 10.1016/j.aei.2025.103375

[15] A. Göbels, F. Rivadeneyra, and J. Beetz, “Transfer of implicit semi-formal textual location descriptions in three-dimensional model contexts”, in *Proceedings of the 2023 European Conference on Computing in Construction and the 40th International CIB W78 Conference*, 40. International CIB W78 Conference, Heraklion (Greece), 10 Jul 2023 - 12 Jul 2023, Jul. 2023. DOI: 10.35490/EC3.2023.268

[16] H. J. Haverkort, “Results on Geometric Networks and Data Structures.”, May 2004.

[17] M. Bonduel, A. Wagner, P. Pauwels, M. Vergauwen, and R. Klein, “Including widespread geometry formats in semantic graphs using RDF literals”, in *EC3 Conference 2019*, ser. Computing in Construction, vol. 1, European Council on Computing in Construction, 2019. DOI: 10.35490/EC3.2019.166

[18] M. Zaharia et al., “Resilient distributed datasets: A {Fault-Tolerant} abstraction for {In-Memory} cluster computing”, in *9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, 2012.