Automation in Construction 193 (2027) 107314

Contents lists available at ScienceDirect

Automation in Construction

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

Ontology-based spatial interlinking of heterogeneous asset data for bridge documentation

Anne Göbels [a,c,*], Oliver Schulz [a,c], Mengyan Peng [b,c], Morris Florek [b,c], Jakob Beetz [a,c], Steffen Marx [b,c], Chongjie Kang [b,c,**]

[a] RWTH Aachen University, Chair of Design Computation, Schinkelstr. 1, Aachen, 52062, Germany
[b] Technical University Dresden, Institute of Concrete Structures, George-Bähr-Str. 1, Dresden, 01069, Germany
[c] Cluster of Excellence CARE, TU Dresden and RWTH Aachen, Germany

ARTICLE INFO

Dataset link: https://doi.org/10.5281/zenodo.23188529, https://github.com/mvg-inatech/pic_2_bridge

Keywords:
Asset management
Knowledge graph
Ontology
Spatial linking
Automation
Legacy data integration
Bridge data

ABSTRACT

Informed decision-making for built assets requires a comprehensive, spatially integrated overview of available documentation. This documentation, however, is typically heterogeneous, siloed, and loosely connected. The spatial relationship of each data source to the structure thereby remains implicit and accessible only to individual experts. To interlink these sources and query them automatically, their spatial relations to the asset must be made machine-readable, yet no vocabulary exists to express them. To address this, the Space Types Ontology (SPOT) is developed to explicitly express these relationships in a format-agnostic way, including the qualitative references that prevail in legacy documentation. The applicability of SPOT is demonstrated by semi-automatically interlinking the documentation of the Nibelungen Bridge in Worms, Germany, into a spatial knowledge graph. This approach makes heterogeneous asset data queryable across data types, reducing dependence on individual expert knowledge and providing a format-independent foundation for informed, data-driven asset management.

1. Introduction

Bridges are critical components of public infrastructure, and their structural integrity must be ensured over their service lives, which often span many decades. In many countries, a large share of this infrastructure is aging and increasingly in critical condition [1,2], which heightens the need for effective maintenance and for a sound information basis for the decisions involved.

These decisions depend on a comprehensive understanding of a bridge's construction, its current condition, and documented history. The data required for this understanding, however, is rarely available in one place. It accumulates over the entire life cycle of the structure, is produced by different authorities and stakeholders across separate planning, construction, and maintenance phases, and varies in format and quality. In addition, a single document can contain information about different aspects of the asset. A construction drawing, for example, often combines several sections, elevations, and details on one sheet, so that distinct sub-spaces of a single file refer to different parts of the structure.

To support reliable decisions, this scattered data must be brought together into a coherent overview. Such consolidation is essentially a matter of understanding how each data item relates to the structure: which aspect or part of the structure is represented or described by a drawing, a photograph, a model, a report, and so on. These spatial references are often only implicit. For example, locations in German bridge documentation are commonly described relative to the structure through directional terms such as front, rear, and left [3], and the position and viewing direction of a 2D drawing are defined by its title and markers. In current practice, understanding and connecting these implicit references of heterogeneous data sources is primarily a manual task that relies on the knowledge of individual experts who have been involved with the structure for long periods. As a result, the process is difficult to scale, ties critical structural knowledge to specific persons, and limits the accessibility of the documented information.

This spatial consolidation challenge is not specific to bridge maintenance data but applies to all life-cycle phases of assets across the built environment, wherever assets are documented by large and heterogeneous data collections.

Thus, this work aims to make the spatial relation of any data item to the structure it documents explicitly machine-readable and storable, so that heterogeneous sources can be integrated into one common

* Corresponding author at: RWTH Aachen University, Chair of Design Computation, Schinkelstr. 1, Aachen, 52062, Germany.
** Corresponding author at: Technical University Dresden, Institute of Concrete Structures, George-Bähr-Str. 1, Dresden, 01069, Germany.
E-mail addresses: goebels@dc.rwth-aachen.de (A. Göbels), chongjie.kang@tu-dresden.de (C. Kang).

https://doi.org/10.1016/j.autcon.2026.107314
Received 16 April 2026; Received in revised form 5 October 2026; Accepted 13 October 2026
Available online 23 October 2026
0926-5805/© 2026 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).
Open access funding enabled and organized by Projekt DEAL.

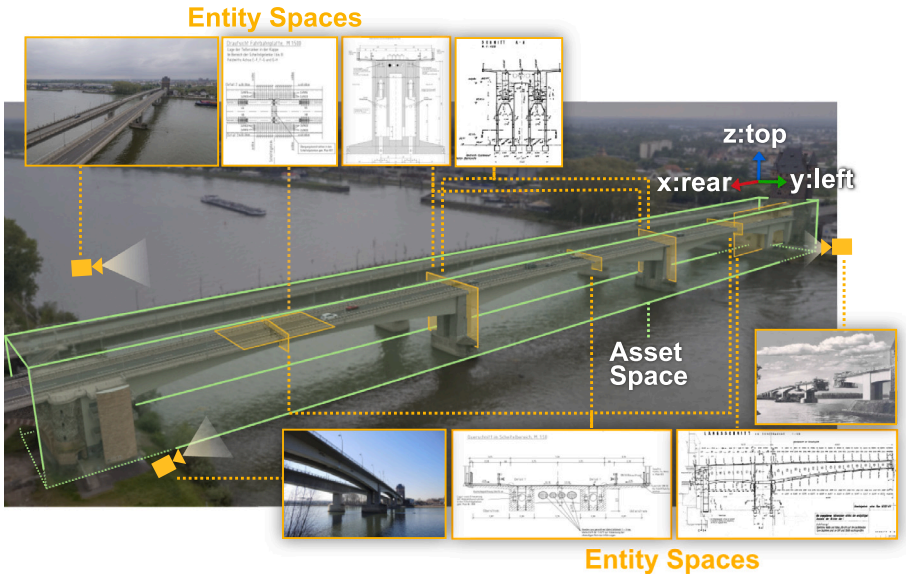


Fig. 1. Concept of spatially linking heterogeneous data to a common asset space.
Image sources: Stefan Gröschel, Prof. Knittel, Chongjie Kang, LBM Worms

spatial context (see Fig. 1), reducing reliance on individual expertise and manual effort. Every data item also carries a temporal reference, which is not considered further here.

Several technical approaches have already addressed parts of this challenge. Common Data Environments (CDEs) [4] store heterogeneous data in their original formats, and container-based approaches such as the Information Container for Linked Document Delivery (ICDD) [5] additionally link these files through explicit, ontology-based links. Such links provide the foundation for semantic integration, but they do not cover spatial aspects. Spatial references can be expressed through coordinate reference systems and geometric transformations in Building Information Modeling (BIM) and Geographic Information System (GIS) contexts. However, these presuppose coordinate-based data and therefore cannot relate arbitrary heterogeneous files to one another. In particular, they do not extend to the non-geometric sources that constitute much of the legacy documentation.

Thus, this work presents an approach to enable such spatial interlinking for any type of data source, with particular emphasis on the implicit, qualitative references that prevail in legacy documentation. Linked Data technology [6] is well suited to this task, matching the decentralized way in which asset data is produced across stakeholders, describing heterogeneous sources through open schemas without a shared fixed format, and providing established ontologies with vocabularies to build on.

In a prior study [7], the spatial characteristics, requirements, and recurring data types of asset documentation were analyzed. Based on these findings, the study defined three central spatial concepts: the *Asset Space* of the physical structure, the *Entity Space* of an individual data item documenting part of it, and the *Document Space* of the corresponding source file.

In this paper, we present a formal ontology of these concepts that defines how their spatial aspects can be described and how different spatial references, especially qualitative ones, can be attached to them in a format-agnostic way. The ontology thereby expresses geometric notions such as boundaries and viewing direction within a semantic vocabulary alongside drafting terms and directional descriptions used in practice. Applying the developed ontology yields a spatial Resource Description Framework (RDF) [8] graph of asset data in which previously disconnected data items are connected through their spatial context, enabling spatial querying and visualization across the dataset.

To assess its practical applicability, the ontology is applied to bridge documentation using a semi-automated pipeline that combines established methods to automatically determine the spatial position of existing photographs and 2D drawings, which represent typical legacy data formats, and integrates them into the spatial knowledge graph. The application is demonstrated on data from the Nibelungen Bridge in Worms, Germany, and serves as a proof of concept for using the ontology with real existing bridge data.

The remainder of this paper is organized as follows. Section 2 reviews related approaches to asset data management and spatial linking, and details the research gap. Section 3 then introduces the developed ontology as the central contribution of this work. Its practical application is subsequently demonstrated in Section 4 on the use-case bridge data. Section 5 presents the results, and Section 6 discusses the ontology and its application. Section 7 finally concludes the paper.

2. Related work

The review is structured around two questions: which methods are used to integrate heterogeneous asset data (Section 2.1), and what spatial information such an integration can carry, that is, which means of representing spatial relations are available to it (Section 2.2). Together, they show what a machine-readable spatial integration of asset data can build on, and where it is still lacking. The focus is on the Architecture, Engineering and Construction (AEC) domain, while the GIS domain is used as a reference point here, because it has standardized RDF-based representations and query functions for spatial data, most prominently through GeoSPARQL [9,10].

2.1. Integration of heterogeneous asset data

For existing built assets, the data landscape is particularly heterogeneous [11]: plans, images, point clouds, and models originate from different periods of time, from different stakeholders, and were created using different tools. Often, the native format is no longer available; instead, only the end product of the processing carried out at the time remains.

A widely used principle in construction information management is to provide a controlled point of access to project information. CDEs, for example, are described as central environments for managing processes and information across the life cycle of a built asset, with information

containers acting as a set of information [12] for files or models. However, such a principle does not necessarily imply that all information must be stored in a single physical repository. It can also be compatible with distributed or federated data storage [13].

Two approaches to integrating heterogeneous data differ in the mechanism by which integration is achieved: through conformance of all data to a shared schema, or through explicit links at a meta-level above the resources, where containers of equal rank are related to one another.

The schema-based approach is followed in BIM and Heritage BIM (HBIM) processes, where the Industry Foundation Classes (IFC) format typically serves as the shared schema. Information of different kinds is expressed directly in this target schema and attached to the elements of the BIM model. Information that cannot be represented in the schema directly can be integrated through placeholder elements, such as *IfcProxyElements*. The shared schema also allows the information to be distributed across several partial models, which are then implicitly superimposed on one another in a coordination process [14]. However, neither this process nor the spatial relation information it establishes is recorded in a machine-readable format. For existing assets, the approach additionally presupposes such a model, which is often not available to begin with [11].

The link-based approach stores heterogeneous resources as information containers: a named, persistent set of information with descriptive metadata [12], and linked to each other via explicit links at a meta-level, rather than simply being grouped within folder hierarchies. The linking concepts also allow for relationships below the container level, i.e., between individual elements of different containers [5]. This principle is implemented both on a local, container-based foundation in the ICDD standard [5,15] and in a federated, web-based manner as described in Werbrouck et al. [16] and Senthilvel et al. [17]. The same need for linking heterogeneous information also appears in the operational phase, for example in digital twins [18,19], where static documentation is combined with dynamic data such as sensor streams.

In neither case are the spatial relations between heterogeneous resources recorded in a machine-readable form: in the schema-based approach they remain implicit in the coordination process, in the link-based approach the link types provided by current vocabularies express association only.

2.2. Representation of spatial relations

Spatial relationships can generally be expressed quantitatively or qualitatively. Two levels must be distinguished:: the representation of the data, i.e., its qualitative or quantitative storage in a database or repository, and the derivation of statements from it through querying and reasoning. The relation between both levels is asymmetrical: qualitative statements can often be derived from quantitative representations, whereas qualitative representations constrain quantitative values without determining them uniquely.

Quantitative spatial representations describe spatial entities by numerical information, e.g., coordinates, geometries, or spatial extents, expressed in absolute or relative Coordinate Reference Systems (CRS). Metric operations, e.g., distances, intersections, or transformations between coordinate systems, then specify or derive the spatial relationships between these entities. Their strength lies in the mathematical precision with which these relationships are represented and evaluated. They are, however, less suited to expressing approximate or vague spatial statements directly [20].

In the GIS domain, GeoSPARQL exemplifies such quantitative representation and derivation [9,10]. It encodes spatial entities as geometry literals, e.g., Well-Known Text (WKT), whose coordinates are interpreted according to an associated reference system.

The AEC domain offers parallel approaches in the Linked Building Data (LBD) context, namely BIMSPARQL [21] and geof3D [22], which likewise rest on quantitative representations. Yet, these query approaches operate on geometry that is already available in a computationally usable coordinate context. They provide functions for deriving spatial relations from geometry, but they do not persistently describe how heterogeneous resources or subspaces with different local CRS relate to one another before such queries can be executed.

The construction of a common reference frame from several local coordinate systems is instead addressed by scene-graph-like structures [23], in which relative coordinate systems are linked together via parent–child relationships. For heterogeneous data, this principle is reflected in Bonduel [24] and Schulz et al. [25]. The former, links building elements to geometry descriptions using the Geometry Metadata Ontology (GOM), together with the Ontology for Managing Geometry (OMG) [26] and the File Ontology for Geometry formats (FOG) [27]. These descriptions are embedded in a framework of interlinked CRS, and affine transformations express the spatial relations between them.

This presupposes that the resources to be linked either provide explicit geometry descriptions or can be related to a CRS by such transformations. For heterogeneous information about existing assets, that precondition often does not hold. Historical drawings, scanned plans, pictures, and reports carry spatial information without an explicit geometry or a stable coordinate system. Distortions, perspective projections, schematic representations, or incomplete documentation artifacts can further prevent the definition of a reliable affine transformation. GOM is thus well suited for linking geometry descriptions and their reference systems, but it does not cover spatial references to arbitrary resources or subspaces within resources that are not available as coordinate-based geometries.

Across these quantitative approaches, spatial references are established at the level of an entire resource or geometry description. For heterogeneous documents, this granularity is too coarse, since a single resource may contain several spatially distinct subspaces with different local reference systems, orientations, or scales. A general mechanism for identifying and linking arbitrary subspaces within a document as spatial entities is therefore missing.

Qualitative descriptions, or Qualitative Spatial Reasoning (QSR), represent spatial relationships by symbolic categories instead of exact numeric values [20,28]. These categories address different aspects of a spatial relationship. Topological relations capture how spatial entities relate to each other, e.g., whether they are disjoint, touching, overlapping, or contained in another and thereby express whether an entity belongs to a space or is its neighbor. Directional relations describe orientation-dependent positions, e.g., left of, right of, or in front of, and thereby refine the position of an entity with respect to that space and its directions. Both are relevant for describing where something is located.

Spatial topology is accordingly a central component of QSR and is formalized using calculi such as the Region Connection Calculus (RCC) [29] or the Egenhofer intersection models [30]. Directional relations are formalized by orientation-based calculi [31] or by qualitative direction calculi that define operations like inverse and composition [32], which allow orientations along a path to be inferred without coordinates.

A further distinction must be made between reasoning and representation in QSR [33]: while spatial reasoning yields qualitative results, these may be based on both qualitative and quantitative representations. In the GIS domain, for example, GeoSPARQL offers topological functions that derive qualitative statements about places and spaces from quantitative geometries. In the AEC domain, topological [34] and directional operators [35] have been discussed for spatial queries and BIMSPARQL and geof3D implement this principle on the quantitative representations described above.

Qualitative representations, by contrast, use statements themselves as a data model and store them persistently in the database. In the AEC Linked Data domain, they remain limited to topological relations. The Building Topology Ontology (BOT) [36] and Bridge Topology Ontology (BROT) [37] cover containment and intersection between

elements, and the Area of Interest Ontology (AOI) [38] covers area-based relations on building elements. None of them capture directional relationships.

One approach that allows directional qualitative representations of spatial entities is the Relative Location Ontology (RELOC) [39]. It combines topological and directional statements to describe the location of an entity relative to a reference entity within a common oriented space. That common space is defined by mapping relative directions, e.g., front, left, or top, to its longitudinal, transversal, and vertical axes. On this basis, the ontology can, express that an entity is located at the front, left side of a reference entity. RELOC defines five topological relationships (disjoint, meet, intersect, contained in, and equal) and three directional properties per axis, e.g., front, longitudinal center, rear for the longitudinal axis. Both are merged into combined properties: 'containedInRear', places an entity in the rear part of the reference entity, 'meetLeft' directly at its left side, and 'disjointTop' above it. RELOC thereby adds the directional dimension to topological statements and enables detailed qualitative representations. Spatial relations between heterogeneous spaces with differing orientations, however, remain outside of its scope.

2.3. Research gaps

Each of the approaches reviewed above covers a different part of the relation between asset data. Existing link vocabularies state which resources, or which of their elements, belong together. Quantitative representations can state where a resource is located, but only if it is available as coordinate-based geometry or can be related to a coordinate system through a transformation. Qualitative representations state topological containment between building elements, and in the case of RELOC also the direction relation between two spaces that share a common orientation.

Given the aim of expressing the spatial relation of any data item to the asset it documents, three aspects remain unresolved by existing approaches. First, no shared vocabulary exists that spatially relates a data item, or an individual subspace of it, to the asset it documents independently of the format in which the item is stored and without requiring its prior conversion into a specific schema such as IFC. Second, resources whose spatial reference is neither geometric nor coordinate-based remain outside any common spatial context, although such sources are widespread in legacy documentation. Third, persistent qualitative representations are confined to topological relations between elements, or to directions within one shared orientation, and therefore cannot express how spaces of differing orientation and extent relate to one another.

3. Space types ontology

To address the gaps presented above, this work develops the Space Types Ontology (SPOT). The ontology leverages the inherent spatial relationship of each data item to the asset it documents as a common property that allows the integration of heterogeneous data within a shared spatial context. Thus, the ontology models every data item, statement, or piece of information as a *Space*: a spatial container that occupies a region and can be positioned relative to other spaces (see Fig. 1). The name *Space Types Ontology* refers to the role of a space within the context of asset documentation, i.e., whether it represents the asset, a source document, or an individual entity. Based on that, the ontology offers means to express the spatial relations between these spaces. Applying the ontology creates a spatial knowledge graph of spatially interlinked asset data.

The development of SPOT builds on a prior preliminary study [7], which analyzed the characteristics of bridge documentation. The findings of that study were used to formulate the design requirements presented in Section 3.1. Additionally, the study proposed an initial conceptual schema of space types. The present work develops this schema into a formal OWL ontology [40] that refines and extends the type hierarchy and adds detailed mechanisms to express spatial relations between entities.

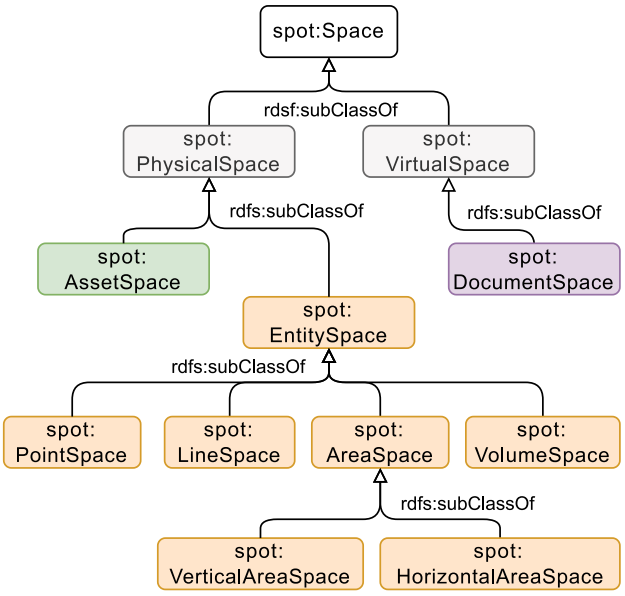


Fig. 2. SPOT space type classes.

As the SPOT ontology should be broadly applicable, it adopts a modular structure that focuses on a minimal set of basic spatial types and simple relations, sufficient for the initial spatial organization of asset data. Where a use case requires more precise definitions, the ontology can be extended or combined with existing ontologies and concepts.

Accordingly, we present the core concepts of SPOT in Section 3.2, followed by its extension, the SPOT-Axis Mapping ontology (SPOT-AM) in Section 3.3, which enables the expression of qualitative references needed to process legacy asset data.

3.1. Ontology design

The design of the SPOT ontology framework follows a set of requirements derived from the gaps identified in Section 2.3 and from the data characteristics analyzed in the prior study [7]. They serve as the specification against which the ontology's structure and functionality are defined:

Requirement 1 (*Req* 1). The ontology should enable the spatial representation of any resource or data related to an asset, regardless of the specific data format or structure.

Requirement 2 (*Req* 2). The ontology should allow the expression of (multiple) spatial relations of a resource to the asset and to its source dataset, and support nested spatial relations.

Requirement 3 (*Req* 3). The ontology should enable the definition of spatial constraints for specific space/resource types that reflect domain standards and rules.

Requirement 4 (*Req* 4). The ontology should support different ways to define spatial references, since existing sources express the location of their content in varying forms. In particular, it should support qualitative, language-based expressions of relations.

Requirement 5 (*Req* 5). The ontology should enable the representation of the orientation and extent of a resource relative to a reference

Table 1
Prefixes used within this paper.

spot:	<https://w3id.org/spot#>
spot-am:	<https://w3id.org/spot/am#>
reloc:	<https://w3id.org/reloc#>
rdf:	<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
rdfs:	<http://www.w3.org/2000/01/rdf-schema#>
owl:	<http://www.w3.org/2002/07/owl#>

space, in order to support targeted queries for spatially complementary information and overlaps between resources.

Requirement 6 (*Req* 6)**.** The ontology should support different levels of certainty for spatial relations, as the reliability of the available locational information differs between sources.

Requirement 7 (*Req* 7)**.** The ontology should have a modular structure that supports domain-specific extensions for specific use cases and focus on the minimum set of concepts required to enable spatial data management.

Based on these design decisions, the ontology framework is intended to answer a set of selected competency questions, following established ontology design principles stated in Grüninger and Fox [41] and Noy and McGuinness [42]. The questions illustrate the desired informative value and query capabilities. They aim primarily at obtaining information about a specific part of the asset from different input details or, conversely, at determining which part of the asset a given resource represents.

Competency Question 1 (*CQ* 1)**.** Which information is available for the front area of the asset?

Competency Question 2 (*CQ* 2)**.** What other resources provide information about the same location of the asset as this resource?

Competency Question 3 (*CQ* 3)**.** What other resources provide spatially complementary information to this resource?

Competency Question 4 (*CQ* 4)**.** Where was this picture taken and in which direction does it view?

Competency Question 5 (*CQ* 5)**.** In which document is this view located, and where on that document?

The ontology is implemented in the OWL Web Ontology Language [40], providing an RDF-based representation of the spatial links and concepts. This enables the format-agnostic linking of heterogeneous, distributed sources and the reuse of and alignment with established AEC domain ontologies (Reqs. 1, 7). Beyond this, the OWL language allows spatial constraints and semantic characteristics to be defined for specific space and relation types (Req. 3), which provides a basis for consistency checking and logical reasoning in future work.

The following sections present the concepts of SPOT and SPOT-AM and show how they fulfill these requirements and support the stated competency questions. Table 1 displays the prefixes used throughout this paper, where ***spot*** is the prefix for the core ontology and ***spot-am*** that of the extension.

3.2. Core concepts

The core concepts of the SPOT ontology comprise a set of basic classes and relations needed to represent the inherent spatial aspects of asset documentation data. Fig. 2 shows the defined space type classes of SPOT and their subclass relations. The core space types are the `Asset Space`, the `Entity Space`, and the `Document Space`. The `Asset Space` represents the space of the actual physical asset whose data is to be managed or documented. It encompasses the entire structure and is the common space to which all data items refer; therefore, it can also be regarded as the highest-level coordinate system.

The `Document Space` represents the space of a source document or file that stores one or more individual `Entity Spaces` (data items). It has an internal independent coordinate system. A typical `Document Space` is, for example, a 2D construction drawing that displays several different views of the structure (e.g., floor plans, cross sections, or elevations). In that case, the size of the `Document Space` corresponds to the extent of the document, for example, in pixel units.

The `Entity Space` represents the space of an individual data item or information statement that documents some part of the asset, for example a technical view, a photograph, or a model. It is usually derived from a `Document Space`, but abstract zones or structural components that do not originate from an external source can also be represented directly by `Entity Spaces`. An `Entity Space` has a local coordinate system, and its size corresponds to the real physical space or region that it represents or describes.

The `Entity Space` has further subclasses that represent different dimensionalities. The `Volume Space` can represent, for example, 3D models or point clouds; the `Area Space` can represent 2D views or sketches; the `Line Space` can depict, for example, water levels; and the `Point Space` can represent the position of a camera or a sensor. The `Area Space` has two subclasses, `Vertical Area Space` and `Horizontal Area Space`, reflecting typical alignments of this space relative to the asset. Thus, in addition to distinguishing resource types, the subclasses enable the definition of spatial constraints for specific entity types (Req. 3).

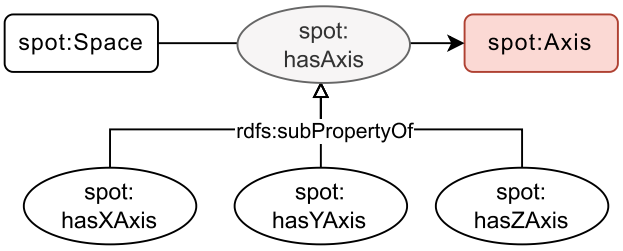
The coordinate system of each SPOT space is defined by axes, which are declared as the x-, y-, or *z*-axis of the space via distinct properties (see Fig. 3(a)). The axes of an `Asset` or `Document Space` are enriched with relative directions (front, right, bottom, etc.) to enable natural-language-based relation definitions and queries (Req. 4) (see Fig. 3(b)).

The definition of the spatial relationship of an `Entity Space` to a reference space is enabled by an `Appearance` object. Through an `Appearance`, an `Entity Space` can be spatially related to the `Asset Space`, a `Document Space`, or another `Entity Space`. Distinct `Appearance` subclasses indicate to which target space an `Entity Space` is linked (see Fig. 3(c)). One `Entity Space` can therefore hold multiple `Appearances`, and even several distinct `Appearances` in the same reference space, for example, when a standard detail drawing applies to several locations in the `Asset Space` (see also Fig. 4). Each `Appearance` thus represents one individual spatial relation of an `Entity Space` to a reference space, and all further details of that relation are attached to it. These details do not have to be defined using the SPOT ontology or an extension of it; they can be specified using any suitable spatial reference mechanism (Req. 4).

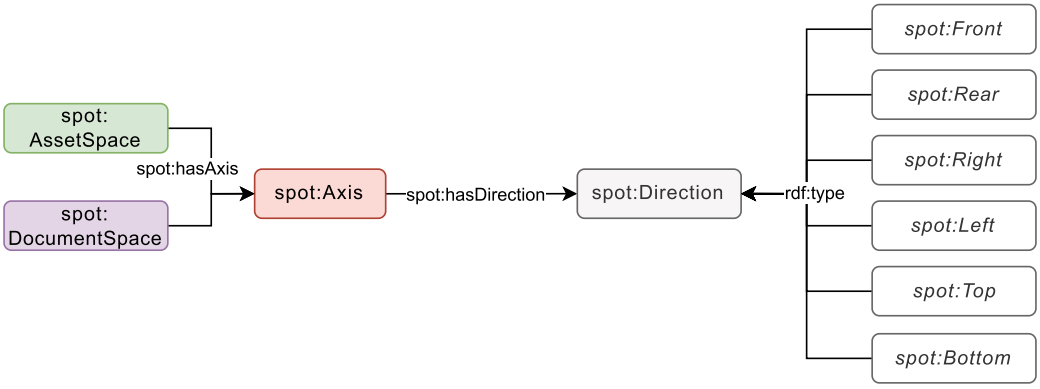
Together, these classes and the `Appearance` mechanism establish the basic structure of SPOT: any data item can be represented as a spatial resource (Req. 1) and related to the asset it describes, to its source document, and/or to another `Entity Space`, which enables also multiple and nested relations (Req. 2).

3.2.1. Application examples of SPOT

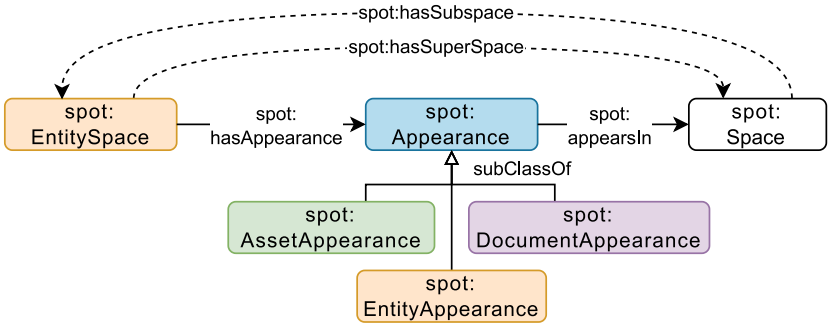
To illustrate how the ontology is applied and which spatial statements can be expressed with the core classes and properties, Fig. 4 shows an example instance graph. In the example, the spatial links established through the `Appearance` objects express that the cross section appears at two locations of the `Asset Space` and originates from an overview drawing. While this does not yet specify where the cross section appears within the reference spaces, the core structure allows various external, existing methods to be integrated to define the spatial reference. The choice of method depends on the available



(a) Space axis definition



(b) Definition of axes directions



(c) Linking of `Entity Spaces` to reference spaces via `Appearances`

Fig. 3. Property definitions of the SPOT ontology.

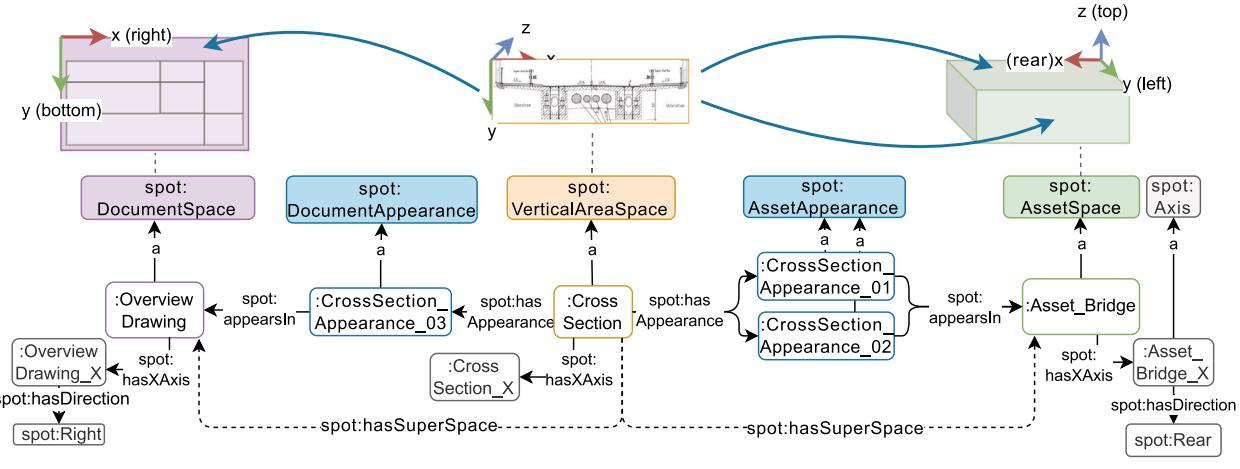
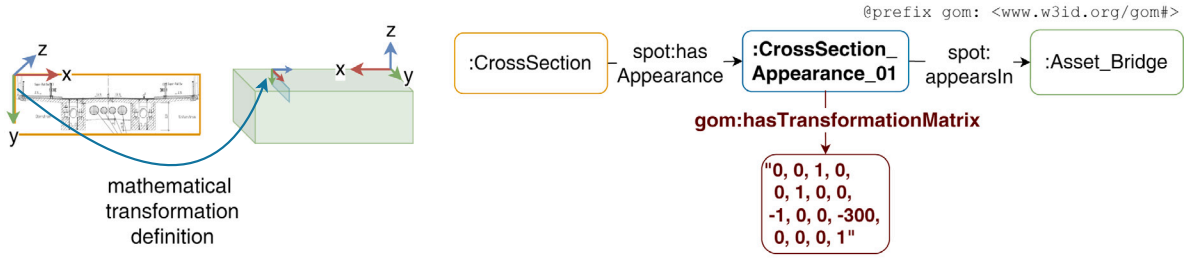
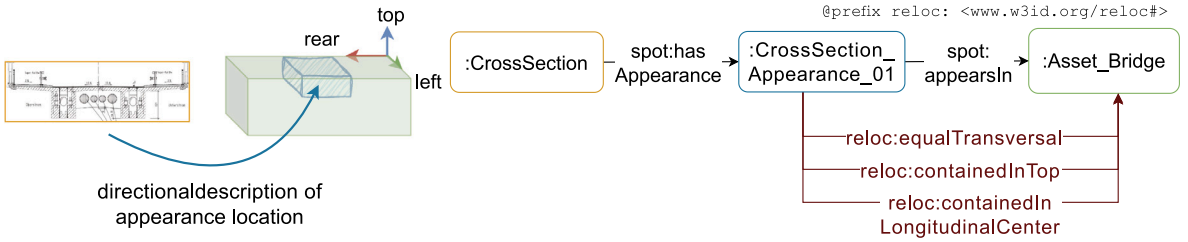


Fig. 4. Space definitions and spatial linking with SPOT core ontology.



(a) Spatial linking with SPOT Core ontology + GOM ontology (exact quantitative spatial reference)



(b) Spatial linking with SPOT core ontology + RELOC ontology (broad qualitative spatial reference)

Fig. 5. Definition of spatial relations using SPOT with external ontologies.

information about the object, its location within the reference spaces, its content type, and the requirements of the specific use case.

If precise spatial positioning is required and sufficient information is available, a transformation matrix can be attached to the `Appearance` object to define the position and rotation of the `Entity Space`'s coordinate system within that of the `Asset Space`. Fig. 5(a) shows how this can be expressed using the GOM ontology [24].

If the exact position of an `Entity Space` is unknown or not needed, a less precise description can be attached. As the axis directions of the `Asset Space` are defined, the position of the entire `Entity Space` can, for example, be described with directional properties of the RELOC ontology, as shown in Fig. 5(b). A purely textual description using an annotation property (comment), or any other qualitative or quantitative method, can equally be used.

3.3. Axis mapping extension

Neither of the two approaches shown in Section 3.2.1 is suitable for spatially interlinking legacy data in a way that explicitly represents the implicit spatial knowledge of experts and enables meaningful spatial queries. Historically collected data often lack the information required for a mathematically precise transformation, while the coarse text-based localization (Fig. 5(b)) provides too few details to determine the orientation of individual `Entity Spaces` relative to one another or the area they cover.

Both aspects are, however, necessary to query the linked dataset for complementary information (Req. 5). The orientation of a resource determines from which direction it depicts the structure. For a 2D drawing, the alignment of its axes encodes the direction from which the view shows the structure, which is a prerequisite for querying complementary or opposing views. For a photograph, the axis alignment corresponds to the viewing direction of the camera and thus indicates which part of the asset is captured. The extent, in turn, defines which region of the asset a resource covers and is required to query overlaps between resources. To capture these aspects, the SPOT ontology is complemented by the Axis Mapping extension (SPOT-AM).

SPOT-AM decomposes a spatial reference into the two parts shown in Fig. 6(b): the orientation, expressed through `Axis Mappings`, and the position and extent, expressed through `Boundary Point Mappings`. Both rely on qualitative expressions, realized by annotating the mappings with descriptive properties.

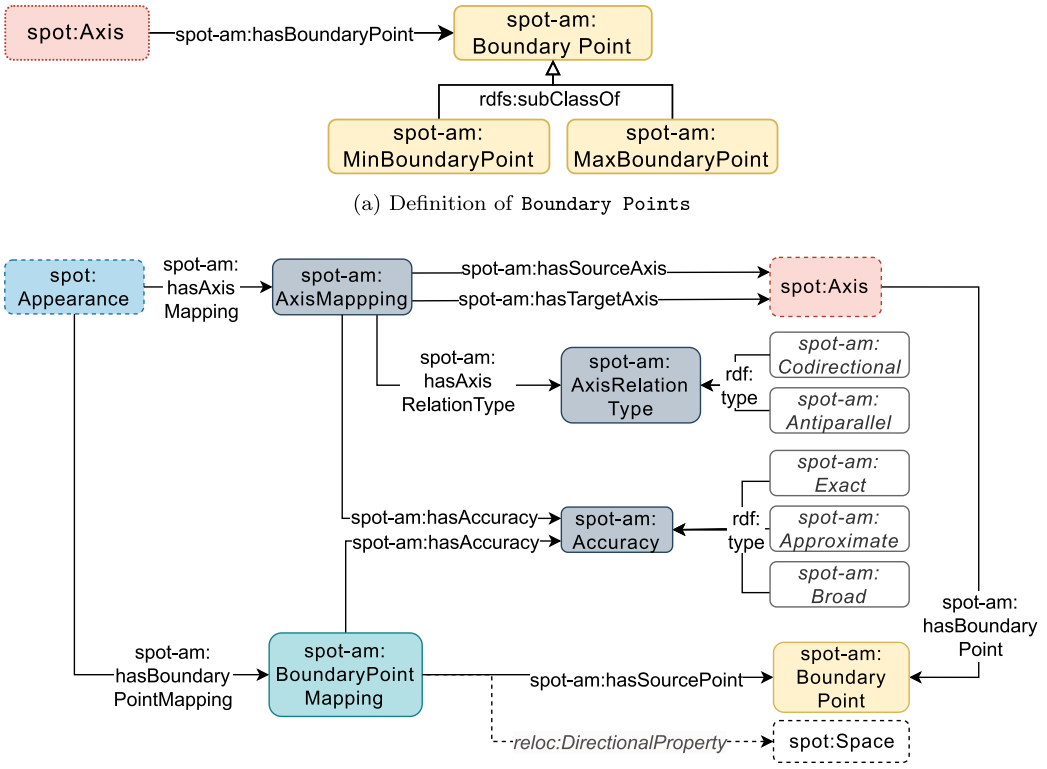
Both mappings are attached to an `Appearance` object of an `Entity Space`. The `Axis Mapping` object links an axis of the `Entity Space` (source axis) to its matching target axis in the respective reference space. The property `spot-am:hasAxisRelationType` describes the directional relation between the two axes: whether they are parallel and point in the same direction (codirectional), or parallel but point in opposite directions (antiparallel). To indicate whether the axes are exactly, approximately, or only broadly parallel, the mapping can be annotated with the `spot-am:hasAccuracy` property (Req. 6).

SPOT-AM further allows `Boundary Points` to be defined for each axis of an `Entity Space`, representing the lower and upper limits of the space along that axis (see Fig. 6(a)). The lower `Boundary Point` (Min) represents the origin of the axis.

The `Boundary Point Mapping` then maps a `Boundary Point` of an `Entity Space` to the reference space. It is defined with directional properties of the RELOC ontology, expressing, for example, that a `Boundary Point` lies on the left side of the reference space, based on the direction of the reference space axis defined with the SPOT core ontology. By defining a minimum and a maximum `Boundary Point` for each axis of an `Entity Space`, not only its position within the reference space but also its extent within that space are represented (Req. 5). A `Boundary Point Mapping` can likewise be annotated with an `Accuracy` value.

This granular qualitative definition of a spatial relation makes it possible to express individual pieces of implicit spatial knowledge explicitly in the knowledge graph, without requiring information that is often unavailable for legacy data, such as the values needed to define a full transformation matrix. For example, a view labeled 'cross section' can be classified as a `Vertical Area Space` with known vertical and transversal axis mappings to the `Asset Space`, while its extent and longitudinal position remain undetermined until further information becomes available.

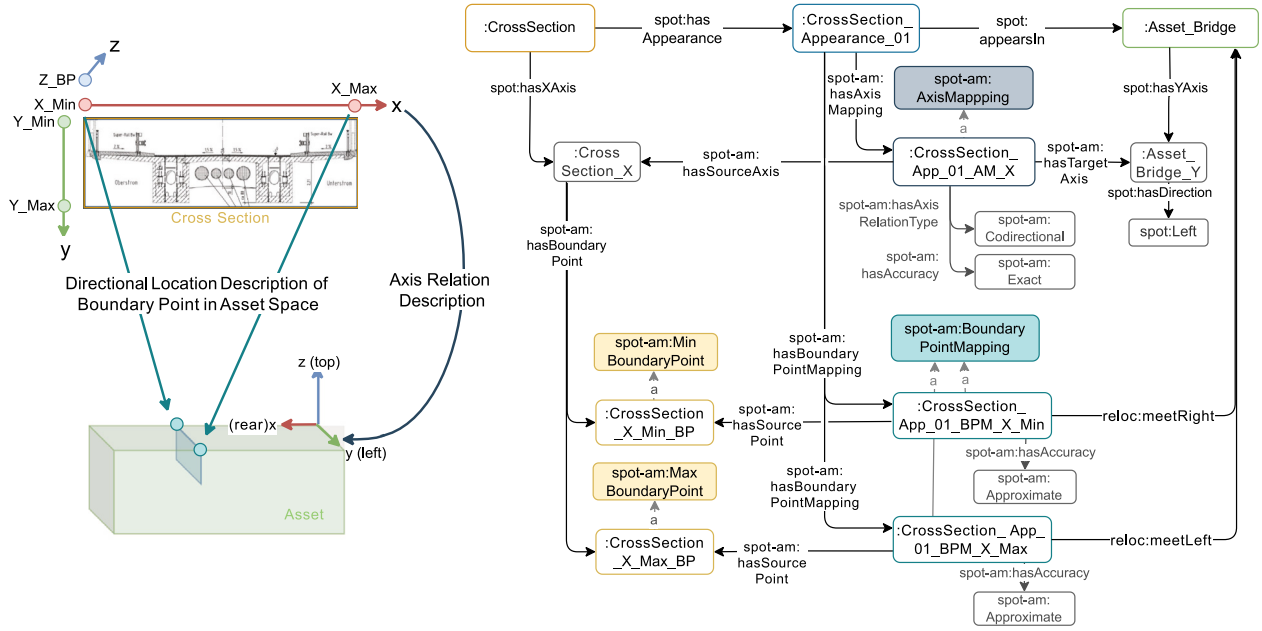
The qualitative directional definition of boundary points is well-suited to representing the inaccuracy or uncertainty of existing data (Req. 6) and is easier to define retrospectively than an exact position. Defining an extent indirectly through a minimum and maximum boundary is also independent of a defined scale of an `Entity Space`'s content, which allows, for example, hand-drawn sketches to be mapped to the `Asset Space`. Such directional references are, moreover, in line



(a) Definition of `Boundary Points`

(b) Definition of `Axis-` and `Boundary Point Mapping`

Fig. 6. Concepts of the axis mapping ontology extension.



Graphic focused on mapping definitions for x-axis of cross section.

Fig. 7. Spatial linking with SPOT core + SPOT-AM ontology.

with the relative descriptions on which bridge documentation already relies and that are accessible for humans to provide and to query.

Fig. 7 shows how the SPOT-AM ontology can be used to add spatial references to the `:CrossSectionAppearance_01` from the example in Fig. 4. The figure focuses on the mappings for the x-axis of the cross section. By defining the mappings for all axes and their boundary points, the position and extent of the cross section within the `Asset Space` are fully specified.

4. Use-case application

To assess its practical applicability, the SPOT ontology framework is applied to real bridge documentation as a proof of concept for

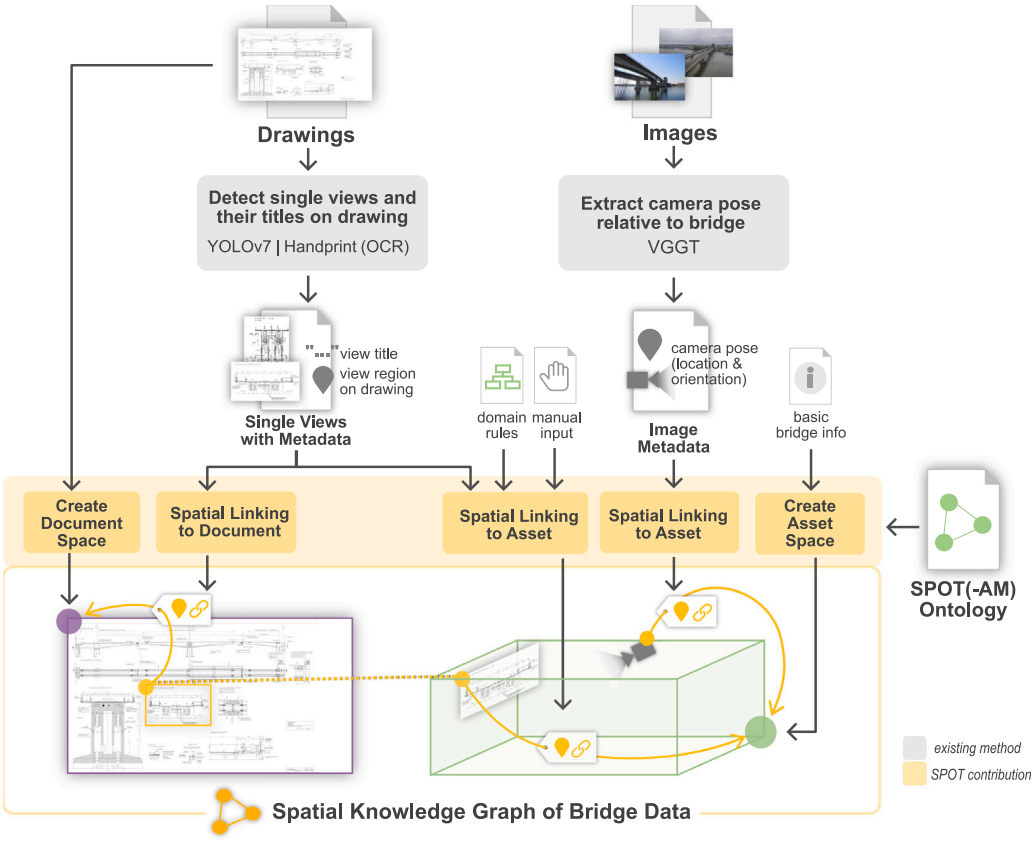


Fig. 8. Semi-automated application process of the SPOT ontology framework for spatially interlinking existing drawing and image data of bridges.

spatially linking and querying heterogeneous legacy bridge data. The demonstration focuses on images and 2D construction drawings, which constitute a large share of the legacy data available for existing assets. The spatial reference of such data to the asset is typically implicit, ambiguous, and of varying quality, so a purely quantitative spatial referencing is often not feasible. Thus, the qualitative, language-based referencing offered by the SPOT-AM extension is used in combination with the SPOT core ontology.

Fig. 8 gives an overview of the application workflow. It starts with the initialization of the Asset Space of the use-case bridge as the common reference space to which all of its data is related (Section 4.1) To automate the demonstration and thereby indicate the practical scalability of the approach, established methods are applied to extract spatial metadata from data that lacks machine-readable spatial references: image-based 3D reconstruction retrospectively recovers the camera position of images depicting the asset (Section 4.2), while Object detection and optical character recognition (OCR) are used to determine the position and title of individual views on combined 2D drawings (Section 4.3). Each of these methods describes the spatial position within its own reference space, that is, the coordinate system of the 3D reconstruction or the individual 2D drawing. The concepts of the SPOT ontology framework are then applied to create a knowledge graph that integrates them into a shared `Asset Space`, thereby enabling spatial linking and querying across image and drawing data at once.

4.1. Asset space creation

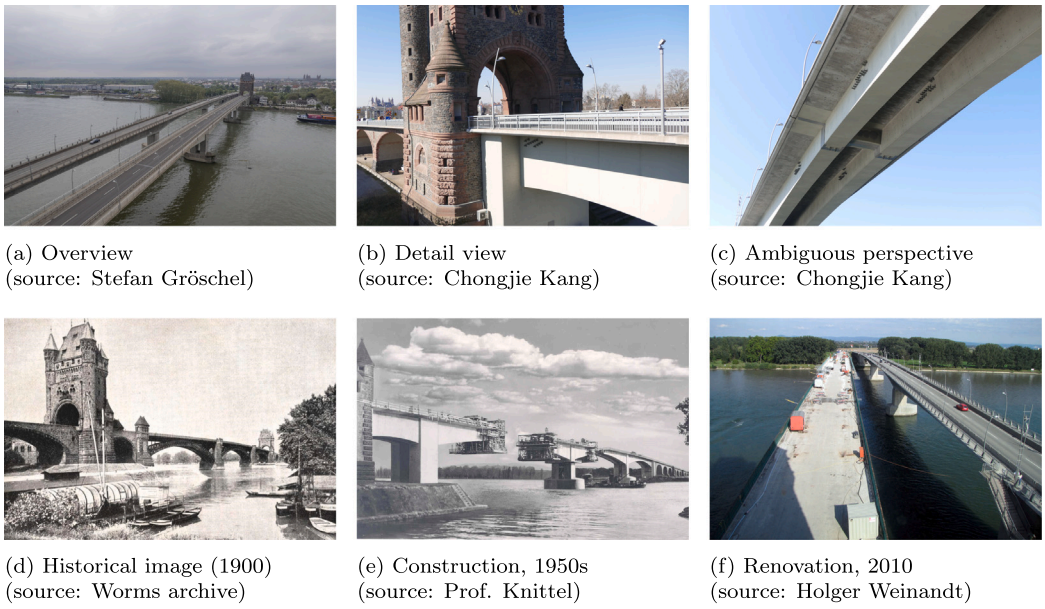
The selected use case is the Nibelungen Bridge in Worms, Germany, shown in Fig. 1. The bridge is 351 m long and divided into four spans. Two considerations motivate this selection. First, the structure is representative of a large part of the existing stock: with its straight alignment, it belongs to the 76% of bridges built in Germany that are straight [43], and its pre-stressed concrete box-girder design is a common bridge type. Bridges in general follow a standardized, well-defined geometry, so reference terms such as ‘cross section of the superstructure’ map to a consistent spatial location regardless of the individual structure, which makes the applied reference logic transferable to many similar assets. Second, the documentation of the bridge reflects the conditions the framework addresses. During its long service life, including reconstruction in 1949 and renovation in the early 2000s, more than 500 drawings and more than 600 photographs have accumulated, differing in provenance, format, and quality. The drawings range from overview and section plans to formwork, reinforcement, and tendon plans as well as detailed views. The composition of this collection was analyzed in the prior study [7].

To generate the spatial knowledge graph, the `Asset Space` is first initialized as the superordinate reference space for all data of the bridge. It is defined by three axes with fixed directions. For this use-case application, the x-axis was defined as the longitudinal axis, pointing toward the rear of the bridge. The orientation was taken from the bridge’s documentation in the national Bridge Maintenance System (BMS). The y-axis is the transversal axis pointing to the left, and the z-axis is the vertical axis pointing upward (see Fig. 1). Additionally, sub-zones of the `Asset Space`, such as super- and substructure, and individual spans, were created as `Volume Spaces` and located within the `Asset Space` to enable more precise localization of data later on (see also Fig. 16).

Listing 1 shows the resulting `Asset Space` definition in the knowledge graph. The directional definition forms the basis for the qualitative localization of the bridge data in the subsequent processing steps.

4.2. Integration of images into the spatial knowledge graph

Correctly interpreting the content of a photograph requires knowing its camera position and orientation. While modern photographs store



(a) Overview (source: Stefan Gröschel)
(b) Detail view (source: Chongjie Kang)
(c) Ambiguous perspective (source: Chongjie Kang)
(d) Historical image (1900) (source: Worms archive)
(e) Construction, 1950s (source: Prof. Knittel)
(f) Renovation, 2010 (source: Holger Weinandt)

Fig. 9. Representative legacy images of the Nibelungen Bridge across categories.

```
:NibelungenBridge a spot:AssetSpace ;
    rdfs:label "Nibelungen Bridge Asset Space"
        ;
    spot:hasSubspace
        :Span_1,:Span_2,:Span_3,:Span_4,
        :SubStructureZone,:SuperStructureZone
            ;
    spot:hasXAxis :NibelungenBridge_Axis_X ;
    spot:hasYAxis :NibelungenBridge_Axis_Y ;
    spot:hasZAxis :NibelungenBridge_Axis_Z .

#Axis Definitions
:NibelungenBridge_Axis_X a spot:Axis ;
    spot:hasDirection spot:Rear .
:NibelungenBridge_Axis_Y a spot:Axis ;
    spot:hasDirection spot:Left .
:NibelungenBridge_Axis_Z a spot:Axis ;
    spot:hasDirection spot:Top .
```

Listing 1: Asset Space definition in the knowledge graph (in turtle syntax)

this information in their metadata, it is often unavailable for legacy images, some of which are decades old. Fig. 9 illustrates the heterogeneity of image data of the Nibelungen Bridge, which is typical for such legacy collections: the examples range from recent overview photographs to detail shots and ambiguous perspectives, as well as historical images of the predecessor bridge (1900), the original construction (1950s), and the most recent renovation (2010–2013). These images differ strongly in perspective, content, and quality, and none of them provides an explicit spatial reference to the structure.

Each image is therefore integrated into the spatial knowledge graph through the position of the camera from which it was taken, represented as a `spot:Point Space` whose z-axis encodes the viewing direction. The camera's position and orientation are linked to the `Asset Space` qualitatively through `Axis-` and `Boundary Point Mappings` of SPOT-AM, which can, e.g., express that an image was taken at the front of the bridge and views toward its rear. This language-based localization matches the way asset experts naturally refer to images and is sufficient to answer spatial queries about them.

To obtain this qualitative localization automatically, image-based 3D reconstruction is leveraged to recover the camera poses of the input images (see Fig. 10). The reconstructed point cloud merely serves as a temporary, intermediate result, as only a broad localization of each image relative to the asset is required. This broad localization is also the only attainable result, since part of the input consists of historical images depicting predecessor states of the bridge, for which the reconstruction cannot form a clean or realistic model of the current structure. The recovered coordinates nevertheless remain sufficient to place each image qualitatively within the `Asset Space`, which keeps the process applicable even to small image sets.

For the reconstruction, the VGGT model [44] is applied to a set of unordered images. For each input image, the VGGT model predicts a dense point map and the extrinsic camera parameters. The per-image point maps are then combined into a single point cloud representing the reconstructed scene, while the extrinsics encode each camera's position and orientation. To handle image sets that exceed the available memory, the reconstruction is performed iteratively: starting from an initial set of images that together cover the entire bridge, additional images are incrementally registered to the scene, so that each newly added image is positioned relative to the initial set. For this application demonstration, five initial and 33 additional images, covering the categories above, were processed.

To express the camera poses with respect to the structure, the reconstructed point cloud has to be aligned with the orientation of the `Asset Space` and the extent of the bridge structure within the point cloud must be identified. To this end, the input images are therefore semantically segmented by bridge components, and the resulting masks are projected into the point cloud for semantic enrichment. This segmentation allows for deriving (broad) boundary coordinates of the bridge structure within the point cloud. However, as the Nibelungen Bridge is in close proximity to a neighboring bridge, which is thus also shown in many pictures, the boundary coordinates have to be manually refined, as the segmentation process cannot separate them automatically. For the alignment, the largest planar surface (e.g., the bridge deck) among the points belonging to the bridge superstructure is identified using the Random Sample Consensus algorithm [45]. The scene is then centered around the plane's centroid, and Principal Component Analysis [46] is applied to the points on the plane to determine the principal directions of the bridge, to which the coordinate axes are aligned.

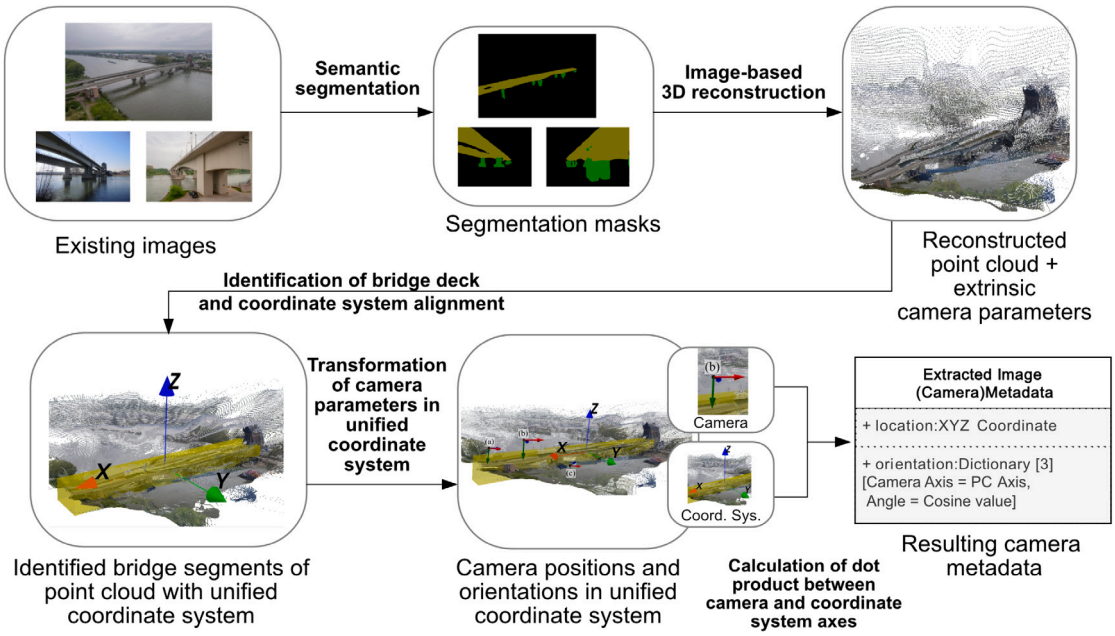
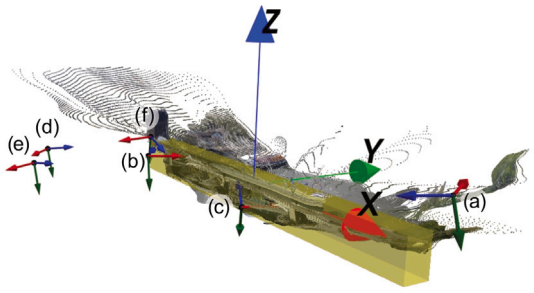


Fig. 10. Process of extracting camera locations and orientations from existing images.



Yellow box marks the bridge extent.

Fig. 11. Aligned point cloud of the Nibelungen Bridge and orientations from the images in Fig. 9.

```
'image_a': {
    'localization': {
        'center':
            [1.14, 0.20, 0.20]
        },
    'orientation': {
        {'x': "y", 'angle':0.91},
        {'y': "z", 'angle':-0.97},
        {'z': "x", 'angle':-0.90}
    }
}
```

Listing 2: Estimated camera location and orientation of Image 9(a), with the center coordinates normalized in relation to the bridge structure.

Finally, the position and orientation of each camera are derived from its extrinsic parameters within this aligned coordinate system. Fig. 11 shows recovered camera positions and orientations for the images in Fig. 9, and Listing 2 displays the metadata record of Image 9(a), containing the camera's center position and the angles between camera axes and their closest match to an axis of the aligned coordinate system, expressed in cosine values.

Based on these recovered poses, each image's camera is added to the graph as a `spot:PointSpace` and spatially linked to the `Nibelungen Bridge Asset Space` through a `spot:AssetAppearance` (see Fig. 12). The position of each Camera Point Space relative to the `Asset Space` is expressed as a qualitative directional relation of the RELOC vocabulary, defined with a `spot-am:BoundaryPointMapping`. The relative relation is derived from the camera's center coordinate in comparison to the identified boundaries of the bridge structure within the point cloud. For the example of Image 9(a), the vertical position of the camera is expressed as `reloc:disjointTop` as the z-coordinate of the camera lies above the upper boundary of the bridge. Since the reconstruction process within this application workflow only delivers approximate results, the mappings are annotated with this accuracy value. The orientation information of the recovered camera metadata is used to create the `spot-am:AxisMappings` between the Camera Point Space and the `Asset Space`. The metadata already defines the target axis of the `Asset Space` for each camera axis. The angle values are used to define the type of axis relation (codirectional or antiparallel) and the `Accuracy` (level of parallelism) of each `Axis Mapping`. For instance, for Image 9(a), the orientation values define that the camera's z-axis is approximately antiparallel to the asset's x-axis.

4.3. Integration of 2D drawings into the spatial knowledge graph

Engineering drawings typically combine several individual technical views of a structure, such as elevations, sections, plan views, and details, on a single sheet. To represent this content in the spatial knowledge graph, each individual view is represented as an `Entity Space` that is linked back to its `Document Space` and to the `Asset Space`.

For this application demonstration, a representative subset of engineering drawings of the Nibelungen Bridge was processed. The subset comprises 24 drawings in total. Of these, 22 are modern CAD-origin drawings produced for the renovation measures carried out between 2010 and 2013 and are available as PDF files with digitally embedded text (see also Fig. 14). In addition, two historical construction drawings from the 1950s were included (see also Fig. 14(b)). These historical drawings are available as scanned raster documents and contain a mixture of printed and handwritten information, as well as visible signs of aging. This combination of modern and historical drawings reflects

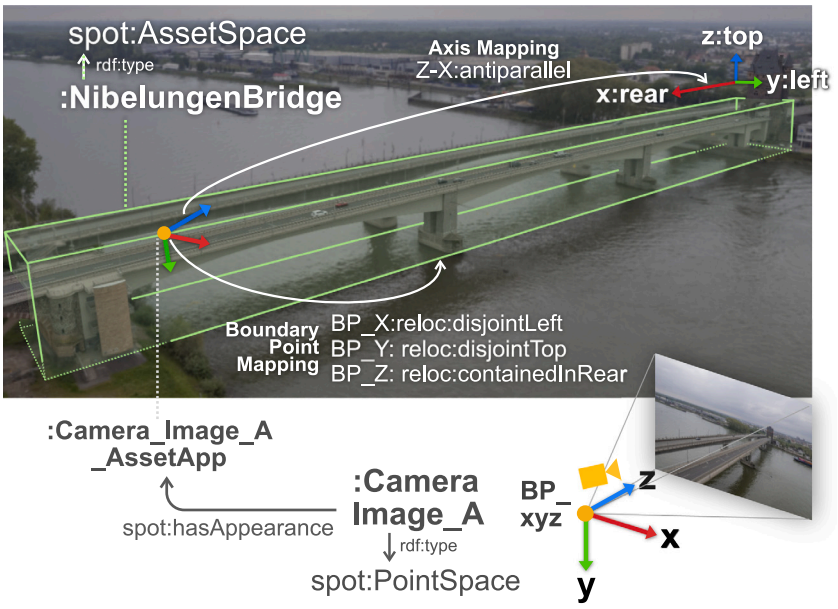


Fig. 12. Resulting knowledge graph: Relation from the camera point space of image 9(a) to the `Asset Space` (simplified).
Image sources: Stefan Gröschel

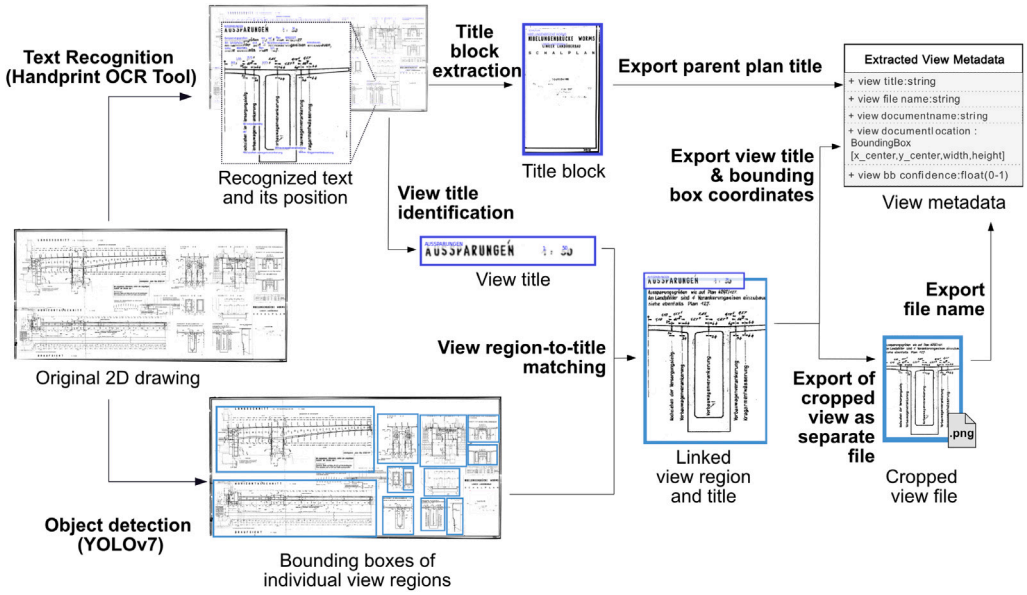
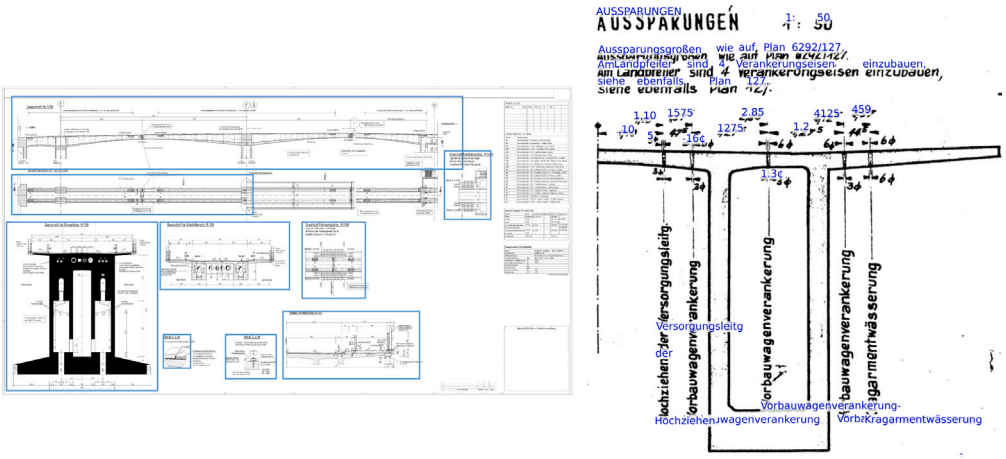


Fig. 13. Process of extracting individual views and textual annotations from 2D drawings.

the heterogeneity typically encountered in legacy bridge documentation and provides a suitable basis for testing the drawing-processing workflow under different data-quality conditions.

To automate the extraction of individual views from a drawing, the complete drawing-analysis pipeline developed by Peng et al. [47] is applied to the Nibelungen Bridge engineering drawings. The pipeline extracts the individual view regions on a drawing alongside the title for each view (see Fig. 13). It combines YOLOv7-based object detection, OCR-based text extraction with the Handprint OCR tool, and spatial post-processing for view-view title association. The object detector first localizes view regions by predicting bounding boxes within the original drawing (see Fig. 14(a)). Each detected view region is then cropped and stored as a separate image while retaining its position within the parent drawing. In parallel, the OCR output provides recognized text elements together with their spatial coordinates (see Fig. 14(b)), enabling the subsequent association of textual and graphical information. The raw object-detection and OCR outputs are further structured through post-processing. First, potential view titles are identified from the OCR-derived text elements using font size information and heuristic filtering rules. Second, the detected view regions are matched with their corresponding view titles based on spatial relations such as containment, intersection, and proximity. This view-view title matching step links each cropped view image to a semantic label and to its original location on the drawing.

For each extracted view, a metadata record is generated (see Listing 3). This record contains the view title, the name of the parent drawing, the normalized bounding box coordinates representing the view region within the original drawing, and the confidence score of the object-detection result. The bounding box coordinates define the vertical and horizontal position and extent expressed in normalized values.



(a) Example of view region localization of a CAD-origin drawing

(b) Example of text recognition of a scanned historical drawing

Fig. 14. Example results of the drawing analysis pipeline.

```
{'view_title': "Querschnitt im
    Scheitelbereich, M. 1:50",
'drawing_title': "Darstellung:
    Bestandsuebersicht Strombruecke Schnitte,
    Details Statik und Konstruktion",
'view_bounding_box':[0.38643, 0.563113,
    0.224048, 0.263174],
'confidence_view_bounding_box': 0.84}
```

Listing 3: Example of view-level metadata exported from the drawing data processing pipeline

To integrate the drawing data into the graph, each drawing is represented as a `spot:DocumentSpace` with a horizontal x-axis and a vertical y-axis, with its origin set to the top-left corner. Based on the extracted metadata, each view of that drawing is subsequently added to the graph as a `spot:AreaSpace` or, if possible, as a `Vertical` or `HorizontalAreaSpace`. This decision is derived from specific terms in the view title, as shown in Fig. 15. For example, if the title contains terms such as 'section' or 'elevation', it usually refers to a vertical view of the asset. An Area Space inherits the vertical and horizontal axis definition and orientation of its `Document Space`, and its z-axis reflects the viewing direction of the view's content.

The relation from a view to its origin `Document Space` is then established via a `spot-am:DocumentAppearance` (see Fig. 16). A View Area Space inherits the axis orientation from its parent `Document Space`; thus, codirectional `Axis Mappings` are defined. The bounding box values of the view's metadata record are used to specify the `Boundary Point Mappings` between the Area Space and the `Document Space`. Since the directions of the `Document Space's` axes are already defined and the bounding box values are normalized, deriving qualitative expressions using the RELOC ontology is straightforward. For instance, x-values in the range of 0 to 0.33 are translated to `reloc:containedInRight`, or y-values in the range of 0.66 to 1 are translated to `reloc:containedInBottom`. In addition, the confidence score of the bounding box is used to annotate the `Boundary Point Mappings` with an Accuracy value.

Next, the `Asset Appearance(s)` of the view are created. Generally, a view can have multiple `Asset Appearances`. Fig. 16 shows, for example, the `Appearance` of the cross section at two different locations within the `Asset Space` as its content is valid for both. The `Axis Mappings` of an `Asset Appearance` can be partially derived from the previously defined class type of the view (see also Fig. 15):

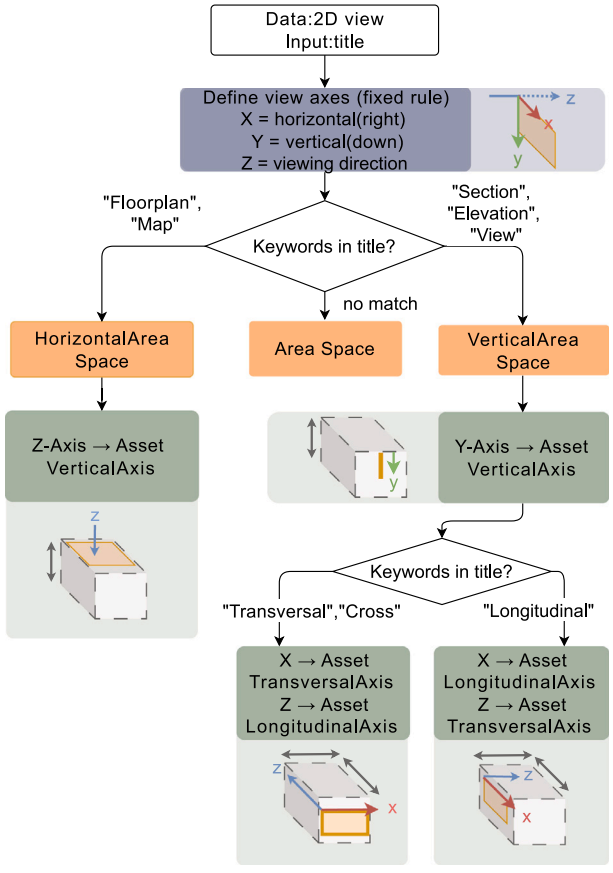


Fig. 15. Rules for deriving `axis mappings` of a view to the asset from a view's title.

if it is a `Horizontal Area Space`, its z-axis (viewing direction) is parallel to the vertical axis of the asset; if it is a `Vertical Area Space`, its z-axis is parallel to the longitudinal or transversal axis of the asset. If possible, more specific `Axis Mappings` are derived from further keywords, such as 'cross' defining that the view's z-axis is parallel to the longitudinal asset axis, or 'longitudinal' meaning that the z-axis is parallel to the transversal asset axis.

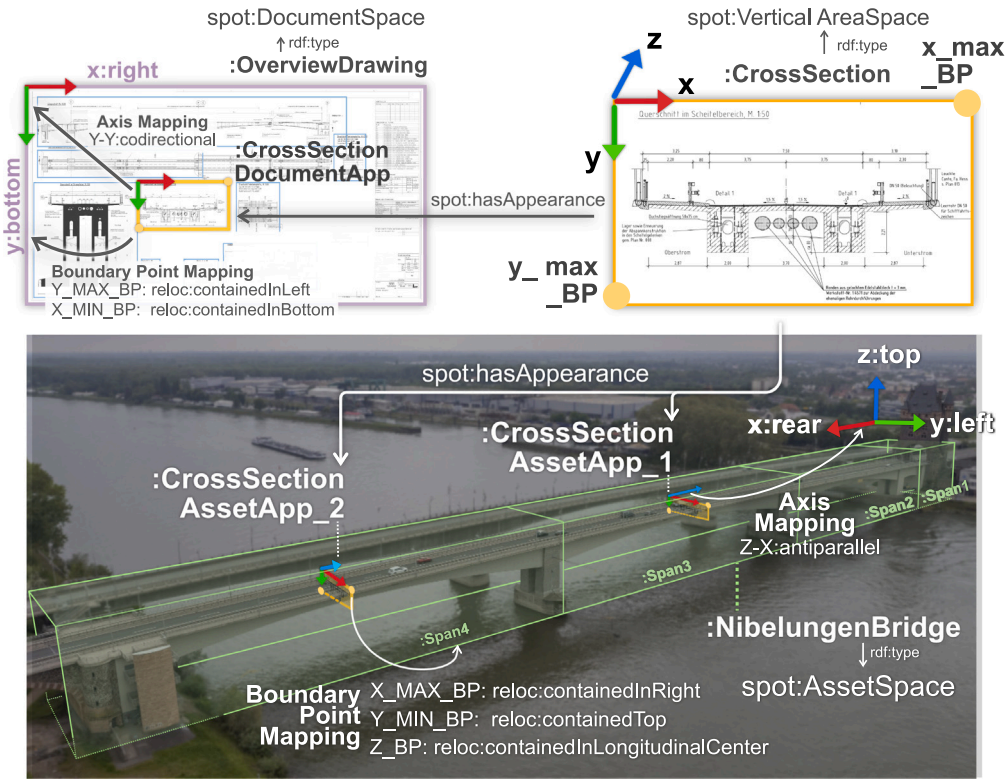


Fig. 16. Resulting knowledge graph: Relations from the cross section area space to the `document` and `Asset Space` (simplified).
Image sources: Stefan Gröschel, LBM Worms

Still, the directional alignment between the mapped axes (pointing in the same or opposite direction) cannot be determined automatically, as it depends on the view's content. Furthermore, the exact position and extent for each Asset `Appearance` of the view within the `Asset Space` cannot be derived from the current processing. Thus, within this prototypical application demonstration, this missing locational data is entered via a simple user interface (see Fig. 17). Using drop-down menus, the `Boundary Point Mappings` of the view's boundary points to the `Asset Space` are defined by selecting a RELOC property. In addition, the missing axis relation types can be added using the interface. Thereby, the `Asset Appearances` of the views are completed, delivering the graph representation displayed in Fig. 16.

5. Results

The application of the SPOT(-AM) ontology to the use-case data delivered a comprehensive spatial knowledge graph. In total, the pipeline processed 62 existing resources and created a graph with 210 individual `Entity Spaces` and their spatial relations. Since the image and drawing processing steps provide the spatial metadata that governs the quality and accuracy of the resulting spatial relations, their intermediate results are briefly reviewed per data type.

For all 38 processed images, comparing the predicted camera positions with the visual content shows that the positions are consistent with the depicted perspectives, and that the orientation is in line with the viewing direction. Even the image of ambiguous perspective (Fig. 9(c)) is reasonably located, as is the historical image (Fig. 9(e)), and also the image depicting the predecessor structure (Fig. 9(d)). For the drawing data, the method extracted 324 individual views from the 24 processed drawings, with 147 having a confidence score above 60% for the reliability of the extracted view region.

The input data, the resulting graph, and the application scripts are openly available,[1] where, due to confidentiality restrictions, the published results cover nine images and two drawings of the use case.

5.1. Competency question queries

The Competency Questions (CQs) 1 to 5 presented in Section 3.1 serve as formal evaluation criteria, assessing whether the ontology fulfills its stated competency requirements. Successfully answering them with SPARQL queries confirms the implementation's internal consistency, thereby verifying that the ontology was built as designed. Applying these queries to the real-world bridge dataset also provides initial validation, demonstrating their applicability to heterogeneous asset documentation data.

The queries to answer the CQs must be able to retrieve various options for spatial references, ranging from simple, direct relations (`Entity Space` → `Asset Space`) to nested relations (`Entity Space` → `Entity Space` → `Asset Space`). As this results in comprehensive query patterns, the listing in this section shows only a snippet of them, while the full queries are presented in Appendix and in the data repository,[2] which also provides the query results for easy review.

For CQ 1 ('Which information is available for the front area of the asset?'), the spatial querying of the graph is initiated by a relative directional keyword. Listing 4 presents the query pattern that searches for all `Entity Spaces` with at least one boundary point mapped

[1] https://doi.org/10.5281/zenodo.23188529, https://github.com/mvg-inatech/pic_2_bridge.
[2] https://doi.org/10.5281/zenodo.23188529.

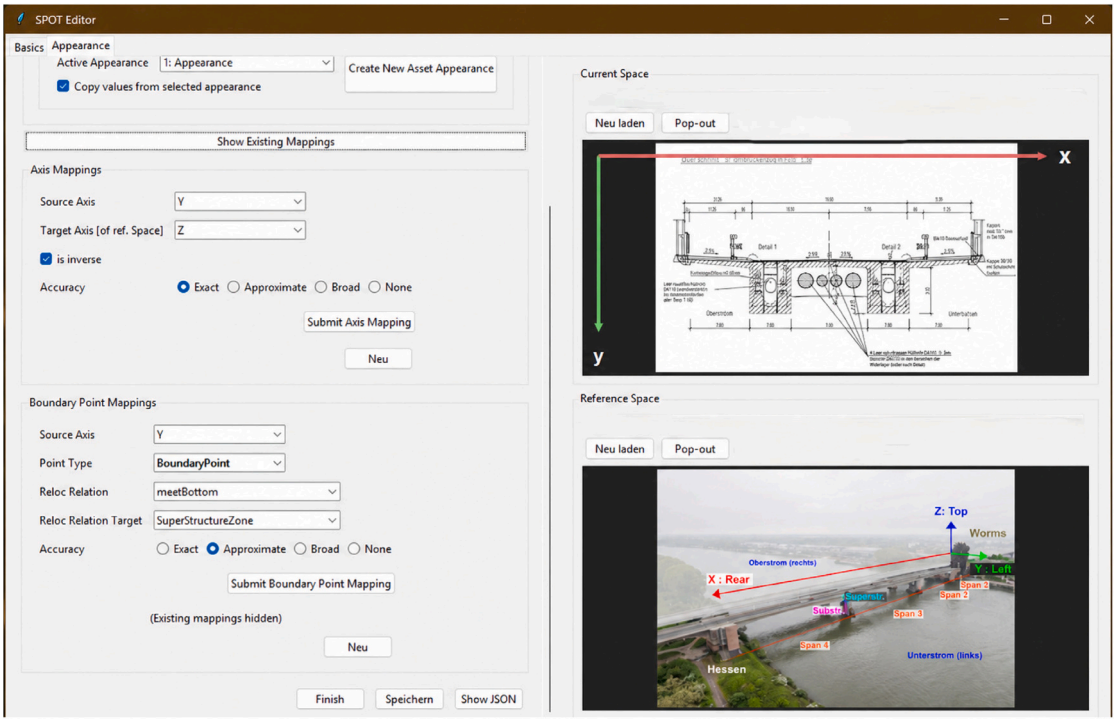


Fig. 17. Simple user interface to define missing `Axis` and `Boundary Point Mapping` specifications.

```
SELECT DISTINCT ?entitySpace
WHERE {
    # Get all entity spaces
    ?entitySpace a/rdfs:subClassOf* spot:
        EntitySpace.
    #get boundary point mappings (bpm) of
        entity space
    ?entitySpace spot:hasAppearance ?appearance
        .
    ?appearance a spot:AssetAppearance,
                spot-am:hasBoundaryPointMapping
                    ?bpm .
    #query for bpms with direct 'front
        relations' to asset space
    ?bpm ?directionalProperty ?asset .
    ?directionalProperty rdfs:subPropertyOf*
        reloc:front.
    }
```

Listing 4: Snippet of SPARQL query for CQ 1

to the front of the asset. The filtering for 'front' is based on the RELOC ontology structure, in which, for example, 'containedInFront' and 'meetFront' are subproperties of 'front'. The full query in Listing 5 also covers resources that are located at the front via nested relationships, thus being somewhere in a reference space that is in turn located at the front part of the asset, or being in the front part of a reference space that has the same longitudinal extent as the `Asset Space`.

CQ 2 ('What other resources provide information about the same location of the asset as 'this resource'?') represents the need to obtain additional or supplementary information about a specific situation. Starting from a selected input space, its location in the `Asset Space` is queried and compared with the locations of all other available resources. The comparison can be performed on the locations expressed via RELOC properties (see Listing 6). The first query returns any spaces that have at least one common spatial relationship with the input space, i.e., the same spatial relationship of a `Boundary Point` to the same reference space, and orders the results by the number of common relationships.

CQ 3 ('What other resources provide spatially complementary information to this resource?') refines CQ 2 (see Listing 7) and filters for resources that show spatially complementary information. For example, if the input for the query is a cross section, the query searches for resources that show the asset from another perspective, such as a longitudinal elevation or a horizontal map. The query is also suited to search for photographs that show the asset from different angles, to achieve a better three-dimensional understanding of the asset.

CQ 4 asks for the location description of a picture's camera position and its viewing direction in relation to the `Asset Space`. The query in Listing 8 illustrates how to search for relations between a camera and the `Asset Space`. The query collects all labels of the relational properties expressing the `Boundary Point Mappings` between the camera and the `Asset Space`, as well as the viewing direction stored in the `Axis Mappings`. The query result for Image 9(c) is, for example: 'image: *:ImgC*; camera location: *is contained in the longitudinal central area of, is contained in the lower area of, right of*; viewing direction: `spot:Front`; direction accuracy: `spot-am:Broad`; asset: *:NibelungenBridge*'.

CQ 5 queries for the relation of an extracted `Entity Space` (view) back to its source `Document Space` (2D drawing). The query shown in Listing 9 returns the drawing's name and file path, as well as the location of the `Entity Space` on it, ensuring easy traceability and provenance management. A result would be: 'View A is contained in the vertical central and upper area, and the transversal central and left area of Plan X'.

Together, the five queries demonstrate that the ontology supports establishing spatial relationships across heterogeneous data types, including qualitative directional properties. This constitutes a functional

evaluation against the stated competency requirements. Executing these queries on a graph populated from real bridge documentation shows that the approach is applicable to existing heterogeneous legacy data and not only to synthetic test cases. The scope of this validation is discussed in Section 6.2.

6. Discussion

The SPOT ontology framework is evaluated against the research gaps identified in Section 2.3 and the design requirements defined in Section 3.1.

6.1. Contributions

The central contribution of this work is the SPOT ontology, which provides the previously missing vocabulary for explicit, machine-readable spatial links between heterogeneous resources. Since the spatial information is held separately from the documentation itself, any data item can be located independently of its format or structure (Req. 1) and related to the asset it documents, to its source document, or to other resources. In the case study, raster photographs as well as scanned analogue and digital drawings are therefore placed in one graph without prior conversion.

Rather than prescribing a single referencing method, the SPOT core represents each resource as a `Space` and attaches its relation to a reference space through an `Appearance` object, to which any suitable referencing method can be bound. In this way, both quantitative references, such as transformation matrices, and qualitative, language-based references can be expressed, without converting all data into a shared schema such as IFC (Req. 4). Since the spatial relation resides in the `Appearance` rather than in the `Space` itself, one entity can hold several such relations at once, including nested ones, both to the asset and to its source (Req. 2). The use-case bridge graph relies on this: a single cross-section drawing is located at several positions of the `Asset Space` through separate `Appearances`, so that one resource can serve as evidence for several parts of the structure. Nesting, in turn, allows data that was captured within another reference space, suchas images taken during point cloud acquisition or annotations made on a drawing, to be located within that space rather than directly on the asset. It is equally required where information is mapped to individual sub-zones or elements of the bridge structure.

Through the distinction between `Document Space` and `Entity Space`, the ontology treats arbitrary subspaces within a document as individual spatial entities, whereas existing scene-graph- and geometry-based approaches reference resources only as a whole. The subclasses additionally allow spatial constraints to be defined for specific resource types (Req. 3). In the case study, assigning a `Vertical` or a `Horizontal Area Space` class to a view already fixes the direction of its z-axis, so that this part of the spatial relation follows from the space type instead of having to be stated for each resource.

The ontology achieves broad applicability through its modular structure: the core is deliberately limited to a minimal set of basic spatial types and relations, so that extensions can supply the more precise definitions a specific use case requires without enlarging the core itself (Req. 7).

The Axis Mapping extension (SPOT-AM) applies this principle: it complements the core by enabling the qualitative, language-based description of the orientation and extent of resources that provide neither precise geometry nor a stable coordinate system (Req. 5), as is typical for legacy documentation. It thereby extends purely directional qualitative approaches such as RELOC, which cannot represent spatial relations between spaces of differing orientation. As the mappings can be annotated with a level of accuracy, the varying quality of legacy metadata is reflected in the graph as well (Req. 6).

Unlike approaches that require a resource's location to be fully determined, SPOT-AM allows that location to be specified incrementally: implicit spatial knowledge held by experts can be made explicit and machine-readable dimension by dimension as it becomes available, and each partial statement is already usable in the graph. The uncertainty inherent in historical records is thereby represented rather than lost.

Taken together, these mechanisms address the three gaps identified in Section 2.3: a format-independent vocabulary relates data items and their subspaces to the asset they document, sources without a geometric reference are placed in a common spatial context, and qualitative relations are no longer confined to topological containment or to a single shared orientation. The framework thereby transforms previously disconnected and implicitly referenced sources into a spatial knowledge graph that can be queried across heterogeneous data types.

For the documentation of the use-case bridge, the competency question queries in Section 5.1 show that this enables spatial questions to be answered directly on legacy data. Retrieving all resources that document a specific part of the asset can, for example, support an on-site inspection: for a crack documented at a given position, the query also returns the section drawings that describe the internal build-up at that location revealing the reinforcement layout and the nominal concrete cover without a manual search through the drawing archive. In the same way, spatially complementary sources can be retrieved that would otherwise have to be identified from individual expert knowledge.

6.2. Limitations

Despite these contributions, both the ontology and its application have limitations that affect the resulting knowledge graph and its query capabilities.

The structure of the SPOT-AM extension introduces graph and query complexity. Spatial relationships between spaces are not modeled as direct links, but are instead described within `Appearance` objects annotated with `Axis` and `Boundary Point Mappings`. This indirection produces large knowledge graphs that are not well suited to direct manual inspection or ad hoc querying, independently of whether the represented relations are nested. At the same time, this indirection is what allows the high degree of freedom in representing data types, dimensionality, and quality described above.

A further limitation concerns the querying of the nested spatial relationships introduced above. Native SPARQL cannot traverse nested relations to arbitrary depth, which fundamentally limits the queries it can express. Accordingly, in CQ 1 the query pattern was explicitly formulated for two levels of nested relationships (input space → reference space → `Asset Space`). Including additional depth levels is possible, but it produces excessively long queries and still cannot handle arbitrary depth in unknown RDF graphs. A further difficulty is the correct higher-level interpretation of nested relative directional terms. For example, an entity located at the rear of a sub-space that itself lies entirely in the front of the asset should also be considered to be in the front of the asset. Currently, this spatial inference must either be hard-coded into the query patterns or resolved manually by the user.

As the `Asset Space` in the use-case graph is subdivided only into broad zones (superstructure, substructure, and spans), a more representative application would additionally require references to individual components, which would further increase the number and depth of nested relationships.

To make the spatial knowledge graph practical and accessible despite this complexity, two complementary approaches can be considered. On the one hand, SPARQL could be extended with spatial query functionalities, in line with BIMSPARQL [21] and GeoSPARQL [9], to query spatial relations directly inside SPARQL. This provides the advantage of including spatial calculation within the query pattern-matching, while carrying the risk of deprecation of the SPARQL extension and limited support from graph database implementations. On the other hand, a hybrid approach that combines SPARQL and client-side computation could use the query to retrieve the spatial data required for the

client to perform the final computation. This approach could use more performant calculations and circumvent the risk of deprecation and limited support, but comes with the disadvantage of not being included in the query's pattern-matching. Investigating how both approaches can be combined remains an open research direction.

A further conceptual limitation of the SPOT ontology is its focus on the position of the data source itself within the `Asset Space`, such as the position of a view or a camera, while disregarding the space that the content of the resource displays, for example, the viewing depth of a 2D section or a photograph. This may lead to incomplete results when querying which sources could provide information about a specific part of the asset. Representing this depicted space is therefore a next step, which would enable more detailed queries and an improved spatial analysis of an asset dataset.

Beyond the space a resource depicts, its individual content is not represented either: the components, dimensions, and annotations shown in a drawing or photograph. For the image data, the segmentation masks created during metadata extraction could, together with the viewpoint and cone of an image, support this object layer by relating detected objects directly to the `Asset Space`. For the drawing data, established methods for recognizing individual components and dimension annotations within a view are already available [48,49] and could be integrated. For the inspection example above, this would allow the concrete cover at the crack position to be queried directly, instead of retrieving the drawing that depicts it.

The application to the Nibelungen Bridge documentation served solely to verify the practical applicability of the ontology, with results sufficient for a proof-of-concept implementation. Nevertheless, the validation of the method can be expanded, as the selection of the use-case dataset currently limits it in terms of the structure's geometry and the available documentation.

The Nibelungen Bridge is straight, so that the axes of the `Asset` and the `Entity Spaces` could be defined orthogonally, and both the definition of the `Asset Space` as the superordinate coordinate system and the definition of the axis mappings were straightforward. As straight bridges account for 76% of the bridges built in Germany [43], the use case addresses a geometry that is representative of a large part of the existing stock. For bridges whose longitudinal axis is curved or skewed, however, the ontology currently lacks the means to express curved axes or non-orthogonal angles within the `Asset Space`. The qualitative relations and the queries used in this work still hold for such structures, as 'left', for instance, remains left relative to the asset regardless of whether the asset itself is curved. As soon as quantitative statements are involved, however, the curvature poses a particular challenge for consistent spatial linking and querying that requires more detailed investigation. In general, the quantitative localization and querying of asset data by means of the SPOT ontology, as well as the combined use of qualitative and quantitative statements, is an important step for future work, since this work focused only on qualitative relations.

For the documentation of the use-case bridge, it was possible to define the required spatial relation of any data item to the asset either automatically or manually. In line with the premise underlying this work, that every data item that documents an asset carries a direct or indirect spatial reference, this precondition is in principle met for all asset data. However, for the legacy data of other assets, it may occur that even with expert knowledge it is not possible to determine the spatial relation of a data item, for example if a photograph shows nothing but a surface or if a drawing is unlabeled or poorly labeled. Such resources can currently only be assigned to the `Asset Space` as a whole. Poorly documented data would, however, equally challenge other data management systems.

The qualitative, directional approach further depends on the definition of an asset direction, which could be derived from the national BMS for the Nibelungen Bridge, as it is part of the mandatory documentation in the German BMS [3]. However, if no such direction is documented and thus, no existing data refers to it, it can also be established once at the beginning of the application process.

A systematic assessment of these preconditions is therefore a worthwhile step for future work, as the reliability with which the spatial reference of a resource can be recovered, and the extent to which the reference information of an asset is documented in other bridge or asset management systems, determine how far the method can be automated and thus applied at scale.

A second constraint on scalability lies in the processing steps themselves. The main limitation of the application workflow is the partially manual localization of the 2D views within the `Asset Space`, as the drawing processing focused on detecting the region and title of each view. The interpretation of view titles as horizontal or vertical relied on simple text comparisons of domain-specific standard terms (section, elevation, floor plan, etc.), which was sufficient for the prototype but should be improved through AI-assisted text interpretation. Detecting the view markers contained in the drawings could enable greater automatic localization [48,49]. The case study is moreover based on a selected subset of drawings, since some could not be processed reliably. The approach performs better on recent drawings than on older historical ones, mainly due to the persistent difficulty of recognizing handwritten text.

The localization of images was fully automated with sufficient results. However, the success of this process is heavily dependent on the manually selected initial input images. The accuracy of the 3D reconstruction process and the detection of the bridge deck for the asset alignment depend on sufficient overlap and varied viewpoints, showing the entire structure from multiple angles. Moreover, the image processing cannot place images that show only detail shots without visible component boundaries.

The use case covered only two data types, images and 2D drawings. They represent the main sources of spatial information for existing assets and already pose high complexity for automatic localization, but integrating further types, such as 3D models, sensors, or videos, remains to be investigated. In previous work, the authors have already shown how textual data from damage documentation can be converted into a spatial representation in the `Asset Space` [50].

Finally, a systematic review by domain experts is still missing, concerning both the conceptual structure of the ontology and the generated graph content. While the competency questions confirm that the ontology can represent and retrieve spatial relationships as designed, an assessment by bridge engineers or asset managers would substantially strengthen the evaluation of both the modeling decisions and the automatically created references, and is identified as a necessary step in future work.

Beyond addressing these limitations, the broader applicability and reuse of the SPOT ontology can be promoted by connecting it with existing LBD domain ontologies, such as the Building [36], Bridge [37], or Damage Topology Ontology [51], which also relate to spatial entities (e.g., components or damages) and their topological relationships. Similarly, an extension of SPOT with geometry and transformation approaches such as GOM [24] and scene-graph-based spatial descriptions [25] can provide a more formal alignment with computational methods for spatial descriptions, compared to the natural-language-aligned approach presented in this work. Beyond the connection and alignment with other ontologies, the deployment of SPOT's spatial knowledge graphs in Digital Twins and container-based environments such as CDEs [4] and ICDD archives [5] is essential to move from the proof of concept demonstrated in this paper toward applicability in operational asset management.

7. Conclusion

This paper has presented the Space Types Ontology (SPOT) as a vocabulary for the machine-readable spatial interlinking of heterogeneous asset data. SPOT enables the explicit expression of the spatial

relation of heterogeneous data, and of individual subspaces within it, to the asset it documents, independently of the format in which it is stored. This relation can then be specified with any suitable referencing method. Through a vendor-agnostic, RDF-based representation, otherwise disconnected data is thereby brought into a comprehensive, interlinked knowledge graph and thus into a common spatial context. Without requiring conversion into a shared format, it enables spatial access and queries across heterogeneous data collections in support of decision-making. Additionally, the Axis Mapping extension SPOT-AM has been presented to address the processing of legacy data, whose spatial references are often only implicit and uncertain. It provides a qualitative, language-based option to define and query the spatial relation between spaces of differing orientation and extent, in line with the approximate, directional way in which locations are commonly expressed in practice.

The practical applicability of the ontology was demonstrated by applying it semi-automatically to the legacy documentation of the Nibelungen Bridge in Worms, Germany. Through the ontology in combination with established methods, existing photographs and 2D construction drawings spanning multiple decades were spatially connected within a knowledge graph. By querying this graph, previously scattered legacy data was spatially accessed across data types and sources, for example to retrieve the resources that document a specific part of the asset or to identify spatially complementary information.

However, the resulting knowledge graph has a complex linking structure, with nested spatial relationships that fit the requirements of heterogeneous asset data but are hard to query, as native SPARQL cannot traverse such nested relations to arbitrary depth. Furthermore, the ontology currently locates the data source itself within the `Asset Space` but not the content it depicts, such as the viewing depth of a section or objects within a photograph, which may leave spatial queries incomplete. Moreover, the use case only demonstrated the use of SPOT with qualitative, directional relations and for a straight bridge geometry with orthogonally definable axes. In addition, the 2D view localization remained partly manual, which currently limits its scalability.

Addressing these points defines the main directions for further research. The query capabilities should be extended with dedicated spatial and traversal functions, and individual objects and the content space of documents should be incorporated into the graph. Beyond this, SPOT should be connected to established Linked Building Data ontologies and tested with further data types, such as models, sensor data, videos, and text, as well as with curved asset geometries and quantitative spatial relations. While the use-case results constitute a proof of concept, systematic expert validation of both the ontology and the generated graph is still required.

The deployment of such spatial knowledge graphs in operational environments such as Common Data Environments and Digital Twins is a key next step toward a spatially integrated basis for decision-making, for example to support and improve the maintenance of bridges. Beyond this, the approach applies to any asset or project of the AEC domain where large and heterogeneous data collections have to be managed, offering a format-independent foundation for spatially interlinking and querying them.

CRediT authorship contribution statement

Anne Göbels: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Investigation, Data curation, Conceptualization. **Oliver Schulz:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Conceptualization. **Mengyan Peng:** Writing – original draft, Visualization, Validation, Methodology, Investigation, Data curation. **Morris Florek:** Writing – review & editing, Writing – original draft, Visualization, Software, Data curation. **Jakob Beetz:** Supervision, Resources, Funding acquisition. **Steffen Marx:** Resources, Funding acquisition. **Chongjie Kang:** Writing – review & editing, Writing – original draft, Validation, Supervision, Funding acquisition, Conceptualization.

Declaration of generative AI and AI-assisted technologies in the manuscript preparation process

During the preparation of this work, the authors used Claude, ChatGPT, and Grammarly solely to check grammar and polish sentences. After using these tools/services, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under the Priority Program SPP 2388 'Hundred plus – Extending the Service Life of Complex Structures through Intelligent Digitalization' (central project number: 501804558, and sub-project numbers: 501812634 and 562882930), the Cluster of Excellence CARE: Climate Neutral and Resource Efficient Construction (EXC 3115, project number: 5337677731), as well as by the German Federal Ministry of Transport through the mFUND funding program and the project HyBridGen (project number: 01FV2067A) and Pic2Bridge (project number: 01F1202A). We also thank the LBM Worms for providing us with the data for the use case.

Appendix. SPARQL queries

Query 1

Query for CQ 1: Which information is available for the *front area* of the asset? (Equivalent queries are possible for other directional zones of the asset)

```
PREFIX : <http://example.org/>
PREFIX spot: <https://w3id.org/spot#>
PREFIX spot-am: <https://w3id.org/spot/am#>
PREFIX reloc: <https://w3id.org/reloc#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dcat: <http://www.w3.org/ns/dcat#>

SELECT DISTINCT ?entitySpace ?relationLabel ?referenceSpace ?path
WHERE {
    # Get all entity spaces
    ?entitySpace a/rdfs:subClassOf* spot:EntitySpace ;
                 dcat:accessURL     ?path .
    #get boundary point mappings (bpms) of entity space
    ?entitySpace spot:hasAppearance ?appearance .
    ?appearance spot-am:hasBoundaryPointMapping ?bpm .
    # OPTION 1: get bpms with direct front relations to asset space
    {
        ?bpm ?directionalProperty ?referenceSpace .
        ?referenceSpace a spot:AssetSpace.
        ?directionalProperty rdfs:subPropertyOf* reloc:front.
    }
    UNION
    # OPTION 2: get bpm with any relation to a reference space, which lies completely in the front area of the asset
    {
        ?bpm ?directionalProperty ?referenceSpace .
        ?referenceSpace a/rdfs:subClassOf* spot:EntitySpace ;
                        spot:hasAppearance ?parentAppearance ;
                        ?hasAxis           ?axis.
        #get longitudinal axis of reference space
        ?axis a spot:Axis .
        ?axismapping spot-am:hasSourceAxis ?axis;
                     spot-am:hasTargetAxis ?targetAxis ;
```

```
                    spot-am:hasAxisRelationType spot-am:Codirectional.
  ?targetAxis spot:hasDirection spot:Rear.
        # get min and max boundary points (BPs) of longitudinal axis
        ?axis spot-am:hasBoundaryPoint ?maxBP, ?minBP .
        ?maxBP a spot-am:MaxBoundaryPoint.
        ?minBP a spot-am:MinBoundaryPoint.
        ?parentAppearance spot-am:hasBoundaryPointMapping ?parentBpmMin, ?parentBpmMax.
        # get min BPs of reference space at the front of the asset or front spans
        ?parentBpmMin spot-am:hasSourcePoint ?minBP;
                      ?parentMinRelType      ?grandParent1 .
        FILTER (?grandParent1 = :NibelungenBridge || ?grandParent1 = :Span_1 || ?grandParent1 = :Span_2)
        ?parentMinRelType rdfs:subPropertyOf* reloc:front.
        #get max BPs of reference space between the front and the center of the asset
        {
            #OPTION a: max BP has front relation to asset space
            {
                ?parentBpmMax spot-am:hasSourcePoint ?maxBP;
                              ?parentMaxRelType ?grandParent2 .
                ?grandParent2 a spot:AssetSpace .
                FILTER (?parentMaxRelType = reloc:containedInLongitudinalCenter ||
                    EXISTS {?parentMaxRelType rdfs:subPropertyOf* reloc:front})
            }
            UNION
            #OPTION b: max BP has any relation to front spans of the asset
            {?parentBpmMax spot-am:hasSourcePoint ?maxBP;
                           ?parentMaxRelType      ?grandParent2 .
                ?parentMaxRelType rdfs:subPropertyOf* reloc:longitudinalAxisRelation.
                FILTER (?grandParent2 = :Span_1 || ?grandParent2 = :Span_2)}
        } }
    UNION
    #OPTION 3 get bpm with front relations to reference space which is at the front of the asset but can have any extension
    {
        ?bpm ?directionalProperty ?referenceSpace .
        ?referenceSpace a/rdfs:subClassOf* spot:EntitySpace .
        FILTER (?directionalProperty = reloc:containedInLongitudinalCenter || EXISTS {?directionalProperty rdfs:subPropertyOf* reloc:front }
            )
        # check that the reference space is also at the front of the asset
        ?referenceSpace spot:hasAppearance ?parentAppearance .
        ?parentAppearance spot-am:hasBoundaryPointMapping ?parentBpm.
        ?parentBpm ?parentRelType ?grandParent .
        ?parentRelType rdfs:subPropertyOf* reloc:front
        FILTER (?grandParent = :NibelungenBridge || ?grandParent = :Span_1 || ?grandParent = :Span_2)
    }
     ?directionalProperty rdfs:label ?relationLabel.
    FILTER langMatches(lang(?relationLabel), "en")
}
```

Listing 5: SPARQL query for CQ 1

Query 2

Query for CQ 2 : Which other resources are available for the same zone of the asset, as resource '*resource-name*'? (resource here: Crosssection 'Aussparungen')

```
PREFIX : <http://example.org/>
PREFIX spot: <https://w3id.org/spot#>
PREFIX reloc: <https://w3id.org/reloc#>
PREFIX spot-am: <https://w3id.org/spot/am#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dcat: <http://www.w3.org/ns/dcat#>

SELECT DISTINCT ?otherEntitySpace (GROUP_CONCAT(DISTINCT (?relationLabel); separator=", ") AS ?dirProps) ?path
WHERE {
  # define variable input via label
  # get appearance (app) /location of variable space in asset
  ?var rdfs:label "AUSSPARUNGEN Area Space";
       spot:hasAppearance ?appVar .
  # get Boundary Point Mappings (bpm) of space in asset
  ?appVar spot:appearsIn ?assetSpace ;
          spot-am:hasBoundaryPointMapping ?bpmVar .
  ?assetSpace a spot:AssetSpace .
  ?bpmVar ?directionalProperty ?refSpace .
  ?directionalProperty rdfs:subPropertyOf* reloc:hasRelativeSpatialRelationTo.
  # get other entity spaces and their asset or entity appearances with their bpms
  ?otherEntitySpace a/rdfs:subClassOf* spot:EntitySpace ;
                    spot:hasAppearance ?appOther ;
       dcat:accessURL ?path .
  ?appOther a ?appCls;
            spot-am:hasBoundaryPointMapping ?bpmOther .
  FILTER(?appCls IN(spot:AssetAppearance, spot:EntityAppearance))
  #filter for other entity spaces with same relation to same reference space
  ?bpmOther ?directionalProperty ?refSpaceOther.
  # refSpaceOther is same as refSpace, or space with same boundaries (asset = trimmed point cloud) (simplified here)
  FILTER ( IF(?refSpace
           IN (:Trimmed_Reconstruction_PointCloud, ?assetSpace),
           ?refSpaceOther IN (?refSpace, ?assetSpace, :Trimmed_Reconstruction_PointCloud),
            ?refSpaceOther = ?refSpace ) )
  FILTER(?otherEntitySpace != ?var)
  ?directionalProperty rdfs:label ?relationLabel.
  FILTER langMatches(lang(?relationLabel), "en")
}
GROUP BY ?var ?otherEntitySpace ?path
# order results by number of same/similar relations
ORDER BY DESC (COUNT(DISTINCT ?relationLabel))
```

Listing 6: SPARQL query for CQ 2

Query 3

Query for CQ 3 : What other resources provide spatially complementary information to 'resource name' ? (resource here: Crosssection)

```

PREFIX : <http://example.org/>
PREFIX spot: <https://w3id.org/spot#>
PREFIX reloc: <https://w3id.org/reloc#>
PREFIX spot-am: <https://w3id.org/spot/am#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dcat: <http://www.w3.org/ns/dcat#>

SELECT DISTINCT ?otherEntitySpace (GROUP_CONCAT(DISTINCT (?relationLabel); separator=", ") AS ?dirProps) ?refSpace ?path
WHERE {
  # define variable input via label
  # get appearance/location of variable space in asset
  ?var rdfs:label "Querschnitt im Scheitelbereich, M_ 1_50 Vertical Area Space";
       spot:hasAppearance ?appVar;
       spot:hasZAxis ?varZ.
  ?appVar a spot:AssetAppearance;
          spot-am:hasBoundaryPointMapping ?bpmVar ;
          spot-am:hasAxisMapping ?amVar.
```

```
  ?bpmVar ?directionalProperty ?refSpace .
  ?directionalProperty rdfs:subPropertyOf* reloc:directionalRelation.
  # get orientation to asset
  ?amVar spot-am:hasSourceAxis ?varZ;
         spot-am:hasTargetAxis ?assetAxis_am_VarZ.
  # get other entity spaces and their asset appearances (bpms)
  ?otherEntitySpace a/rdfs:subClassOf* spot:EntitySpace ;
                    spot:hasAppearance ?appOther ;
                    spot:hasZAxis ?otherZ;
        dcat:accessURL ?path .
  ?appOther a spot:AssetAppearance;
            spot-am:hasBoundaryPointMapping ?bpmOther ;
            spot-am:hasAxisMapping ?amOther.
  #filter for other entity spaces with same relation to same reference space
  ?bpmOther ?directionalProperty ?refSpaceOther.
  FILTER ( ?refSpaceOther = ?refSpace )
  #filter other entity spaces with complimentary orientations
  ?amOther spot-am:hasSourceAxis ?otherZ;
         spot-am:hasTargetAxis ?assetAxis_am_OtherZ.
  FILTER(?assetAxis_am_OtherZ != ?assetAxis_am_VarZ)
  FILTER(?otherEntitySpace != ?var)
  ?directionalProperty rdfs:label ?relationLabel.
  FILTER langMatches(lang(?relationLabel), "en")
}
GROUP BY ?var ?otherEntitySpace ?assetAxis_am_VarZ ?assetAxis_am_OtherZ ?refSpace ?path
# order results by number of same/similar relations
ORDER BY DESC (COUNT(DISTINCT ?relationLabel))
```

Listing 7: SPARQL query for CQ 3 - based on directional properties

Query 4

Query for CQ 4 : Where was *this picture* taken and in which direction does it view?

```

PREFIX : <http://example.org/>
PREFIX spot: <https://w3id.org/spot#>
PREFIX reloc: <https://w3id.org/reloc#>
PREFIX spot-am: <https://w3id.org/spot/am#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dcat: <http://www.w3.org/ns/dcat#>
PREFIX dcterms: <http://purl.org/dc/terms/>

SELECT ?img
       (GROUP_CONCAT(DISTINCT ?relationLabel; separator=", ") AS ?cameraLocation) ?viewingDirection ?viewingDirectionAccuracy
       ?refSpace ?path
WHERE {
  ?img a spot:PointSpace ;
     dcterms:type "image";
     dcat:accessURL ?path ;
     # zaxis indicating viewing direction
     spot:hasZAxis ?imgZAxis;
     spot:hasAppearance ?appImg .
     ?appImg spot:appearsIn ?refSpace ;
          spot-am:hasBoundaryPointMapping ?bpmImg ;
          spot-am:hasAxisMapping ?amImg.
    ?bpmImg ?directionalProperty ?refSpace .
    ?directionalProperty rdfs:subPropertyOf* reloc:directionalRelation;
                         rdfs:label ?relationLabel.
     FILTER langMatches(lang(?relationLabel), "en")
    ?amImg spot-am:hasSourceAxis ?imgZAxis;
           spot-am:hasTargetAxis ?refSpaceAxis;
           spot-am:hasAxisRelationType ?relType;
           spot-am:hasAccuracy ?viewingDirectionAccuracy.
    ?refSpaceAxis spot:hasDirection ?refSpaceAxisDir.
    ?refSpaceAxisDir spot:isInverseDirectionOf ?refSpaceAxisInvDir.
    BIND(
      IF(?relType = spot-am:Antiparallel, ?refSpaceAxisInvDir, ?refSpaceAxisDir) AS ?viewingDirection)
}
GROUP BY ?img ?viewingDirection ?viewingDirectionAccuracy ?refSpace ?path
ORDER BY ?img
```

Listing 8: SPARQL query for CQ 4

Query 5

Query for CQ 5 : Where (which document, where on this document) can *this drawing* be found in the documents?

```
PREFIX : <http://example.org/>
PREFIX spot: <https://w3id.org/spot#>
PREFIX reloc: <https://w3id.org/reloc#>
PREFIX spot-am: <https://w3id.org/spot/am#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dcat: <http://www.w3.org/ns/dcat#>

SELECT DISTINCT ?var ?relationLabel ?refSpace ?refSpaceDesc ?path
WHERE {
  #define variable input via label
  ?var rdfs:label "AUSSPARUNGEN Area Space";
       spot:hasAppearance ?appVar .
  ?appVar a spot:DocumentAppearance .
  # get location of variable space on document space
  ?appVar spot-am:hasBoundaryPointMapping ?bpmVar .
  ?bpmVar ?directionalProperty ?refSpace.
  ?directionalProperty rdfs:subPropertyOf* reloc:directionalRelation;
                       rdfs:label ?relationLabel.
  FILTER langMatches(lang(?relationLabel), "en")
  # get name and path of document space
  ?refSpace dcat:accessURL ?path ;
            rdfs:comment ?refSpaceDesc.
}
```

Listing 9: SPARQL query for CQ 5

Data availability

Selected examples of the use-case input data, the source code for graph creation, and the method results are available in an online repository (https://doi.org/10.5281/zenodo.23188529). The image data processing codebase is publicly available on GitHub (https://github.com/mvg-inatech/pic_2_bridge).

References

[1] American Society of Civil Engineers, ASCE's 2021 Infrastructure Report Card, Technical Report, American Society of Civil Engineers, 2021, (Accessed 10 January 2025). URL: https://infrastructurereportcard.org/wp-content/uploads/2020/12/Bridges-2021.pdf.

[2] Bundesanstalt für Straßenwesen, Brueckenstatistik, 2026, (Accessed 26 March 2026). URL: https://www.bast.de/DE/Publikationen/Statistik/Bruecken/Brueckenstatistik.pdf?_blob=publicationFile&v=7. (Accessed 26 March 2026).

[3] Bundesministerium für Digitales und Verkehr, Anweisung straßeninformationsbank - Teilsystem bauwerksdaten 2.05, 2024, (Accessed 19 November 2025). URL: https://www.bast.de/DE/Publikationen/Regelwerke/Verkehrstechnik/Downloads/B01a-Bauwerke.pdf?_blob=publicationFile&v=1.

[4] C. Preidel, A. Borrmann, H. Mattern, M. König, S.-E. Schapke, Common data environment, in: A. Borrmann, M. König, C. Koch, J. Beetz (Eds.), Building Information Modeling: Technology Foundations and Industry Practice, Springer International Publishing, ISBN: 978-3-319-92862-3, 2018, pp. 279–291, http://dx.doi.org/10.1007/978-3-319-92862-3_15.

[5] DIN Standards Committee Building and Civil Engineering, Information container for linked document delivery exchange specificaton part 1: Container (ISO 21597-1:2020); German version EN ISO 21597-1:2020, 2021, http://dx.doi.org/10.31030/3137795.

[6] P. Pauwels, S. Zhang, Y.-C. Lee, Semantic web technologies in AEC industry: A literature overview, Autom. Constr. 73 (2017) 145–165, http://dx.doi.org/10.1016/j.autcon.2016.10.003.

[7] A. Göbels, O. Schulz, 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, (Accessed 10 October 2024). URL: https://itc.scix.net/paper/w78-2024-13.

[8] W3C RDF Working Group, RDF 1.1 concepts and abstract syntax, 2014, (Accessed 29 February 2024). URL: https://www.w3.org/TR/rdf11-concepts/.

[9] R. Battle, D. Kolas, Enabling the geospatial semantic web with parliament and GeoSPARQL, Semant. Web 3 (4) (2012) 355–370, http://dx.doi.org/10.3233/SW-2012-0065.

[10] N.J. Car, T. Homburg, GeoSPARQL 1.1: An almost decadal update to the most important geospatial LOD standard, in: B. Yaman, M.A. Sherif, A.-C. Ngonga Ngomo, A. Haller (Eds.), Proceedings of the 4th International Workshop on Geospatial Linked Data, GeoLD 2021, Hersonissos, Greece, 2021, pp. 26–33, (Accessed 17 March 2026). URL: https://ceur-ws.org/Vol-2977/paper4.pdf.

[11] R. Volk, J. Stengel, F. Schultmann, Building information modeling (BIM) for existing buildings — Literature review and future needs, Autom. Constr. 38 (2014) 109–127, http://dx.doi.org/10.1016/j.autcon.2013.10.023.

[12] ISO, ISO 19650-1:2018 organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) - information management using building information modelling - Part 1: concepts and principles, 2018, (Accessed 28 April 2020). URL: https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/06/80/68078.html.

[13] J. Werbrouck, O. Schulz, J. Oraskari, E. Mannens, P. Pauwels, J. Beetz, A generic framework for federated CDEs applied to issue management, Adv. Eng. Inform. 58 (2023) 102136, http://dx.doi.org/10.1016/j.aei.2023.102136.

[14] A. Borrmann, M. König, C. Koch, J. Beetz, Building information modeling: Why? What? How? in: Building Information Modeling: Technology Foundations and Industry Practice, Springer International Publishing, ISBN: 978-3-319-92861-6, 2018, pp. 1–24, http://dx.doi.org/10.1007/978-3-319-92862-3.

[15] P. Hagedorn, M. Senthilvel, H. Schevers, L.B. Verhelst, Towards usable ICDD containers for ontology-driven data linking and link validation, in: Proceedings of the 11th Linked Data in Architecture and Construction Workshop, Matera, Italy, 2023, pp. 35–46, (Accessed 17 March 2026). URL: https://ceur-ws.org/Vol-3633/paper3.pdf.

[16] J. Werbrouck, P. Pauwels, J. Beetz, R. Verborgh, E. Mannens, ConSolid: A federated ecosystem for heterogeneous multi-stakeholder projects, Semant. Web 15 (2) (2024) 429–460, http://dx.doi.org/10.3233/SW-233396.

[17] M. Senthilvel, J. Oraskari, J. Beetz, Common data environments for the information container for linked document delivery, in: Proceedings of the 8th Linked Data in Architecture and Construction Workshop, Dublin, Ireland (virtually hosted), 2020, pp. 132–145, URL: https://ceur-ws.org/Vol-2636/10paper.pdf.

[18] M. Grieves, J. Vickers, Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems, in: F.-J. Kahlen, S. Flumerfelt, A. Alves (Eds.), Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches, Springer International Publishing, Cham, ISBN: 978-3-319-38756-7, 2017, pp. 85–113, http://dx.doi.org/10.1007/978-3-319-38756-7_4.

[19] C. Kang, M. Walker, J.-H. Bartels, G. Marzahn, S. Marx, Digital twin technologies for bridge lifecycle management—Literature insights and a pilot study on the nibelungen bridge, Results Eng. 28 (2025) 108288, http://dx.doi.org/10.1016/j.rineng.2025.108288.

[20] A. Cohn, S. Hazarika, Qualitative spatial representation and reasoning: An overview, Fund. Inform. 46 (1–2) (2001) 1–29, http://dx.doi.org/10.3233/FUN-2001-461-202.

[21] C. Zhang, J. Beetz, B. de Vries, BimSPARQL: Domain-specific functional SPARQL extensions for querying RDF building data, Semant. Web 9 (6) (2018) 829–855, http://dx.doi.org/10.3233/SW-180297.

[22] D. Elshani, D. Hernandez, A. Nakhaee, A.A. Arrascue, S. Staab, T. Wortmann, geof3D: SPARQL geometric functions for co-designing buildings, Adv. Eng. Inform. 71 (2026) 104261, http://dx.doi.org/10.1016/j.aei.2025.104261.

[23] S. Marschner, P. Shirley, Data structures for graphics, in: Fundamentals of Computer Graphics, Vol. 4, CRC Press, ISBN: 978-1-4822-2941-7, 2016, pp. 281–335.

[24] M. Bonduel, A Framework for a Linked Data-Based Heritage BIM (Ph.D. thesis), KU Leuven, 2021, (Accessed 16 November 2021). URL: https://lirias.kuleuven.be/retrieve/618662.

[25] O. Schulz, J. Werbrouck, J. Beetz, Towards scene graph descriptions for spatial representations in the built environment, in: Proceedings of the 30th International Workshop on Intelligent Computing in Engineering, EG-ICE 2023, London, United Kingdom, 2023, pp. 1–10, URL: https://www.ucl.ac.uk/bartlett/sites/bartlett/files/towards_scene_graph_descriptions_for_spatial_representations_in_the_built_environment.pdf.

[26] A. Wagner, M. Bonduel, P. Pauwels, R. Uwe, Relating geometry descriptions to its derivatives on the web, in: Proceedings of the 2019 European Conference for Computing in Construction, ISBN: 978-1-910963-37-1, 2019, pp. 304–313, http://dx.doi.org/10.35490/ec3.2019.146.

[27] M. Bonduel, A. Wagner, P. Pauwels, M. Vergauwen, R. Klein, Including widespread geometry formats in semantic graphs using RDF literals, in: Proceedings of the 2019 European Conference on Computing in Construction, 2019, p. 350, http://dx.doi.org/10.35490/EC3.2019.166.

[28] A.G. Cohn, B. Bennett, J. Gooday, N.M. Gotts, Qualitative spatial representation and reasoning with the region connection calculus, GeoInformatica 1 (3) (1997) 275–316, http://dx.doi.org/10.1023/A:1009712514511.

[29] D.A. Randell, Z. Cui, A.G. Cohn, A spatial logic based on regions and connection, in: Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning, KR '92, Cambridge, MA, ISBN: 1558602623, 1992, pp. 165—-176.

[30] M.J. Egenhofer, R.D. Franzosa, Point-set topological spatial relations, Int. J. Geogr. Inf. Syst. 5 (1991) 161–174, http://dx.doi.org/10.1080/02693799108927841.

[31] C. Freksa, Using orientation information for qualitative spatial reasoning, in: Theories and Methods of Spatio-Temporal Reasoning in Geographic Space, Pisa, ISBN: 978-3-540-47333-6, 1992, pp. 162–178, http://dx.doi.org/10.1007/3-540-55966-3_10.

[32] A.U. Frank, Qualitative spatial reasoning: Cardinal directions as an example, Int. J. Geogr. Inf. Syst. 10 (3) (1996) 269–290, http://dx.doi.org/10.1080/02693799608902079.

[33] J. Chen, A.G. Cohn, D. Liu, S. Wang, J. Ouyang, Q. Yu, A survey of qualitative spatial representations, Knowl. Eng. Rev. 30 (1) (2015) 106–136, http://dx.doi.org/10.1017/S0269888913000350.

[34] A. Borrmann, J. Beetz, Towards spatial reasoning on building information models, in: Proceedings of the 8th European Conference on Product and Process Modelling, ECPPM, Cork, Ireland, 2010, pp. 1–6, (Accessed 28 April 2020). URL: https://research.tue.nl/en/publications/towards-spatial-reasoning-on-building-information-models.

[35] A. Borrmann, E. Rank, Specification and implementation of directional operators in a 3D spatial query language for building information models, Adv. Eng. Inform. 23 (1) (2009) 32–44, http://dx.doi.org/10.1016/j.aei.2008.06.005.

[36] M.H. Rasmussen, M. Lefrançois, G.F. Schneider, P. Pauwels, BOT: The building topology ontology of the W3C linked building data group, Semant. Web 12 (1) (2020) 143–161, http://dx.doi.org/10.3233/SW-200385.

[37] A.-H. Hamdan, R.J. Scherer, Integration of BIM-related bridge information in an ontological knowledgebase, in: M. Poveda-Villalón, A. Roxin, K. McGlinn, P. Pauwels (Eds.), Proceedings of the 8th Linked Data in Architecture and Construction Workshop, Dublin, Ireland, 2020, pp. 77–90, URL: https://ceur-ws.org/Vol-2636/06paper.pdf.

[38] A.-H. Hamdan, R.J. Scherer, Areas of interest - semantic description of component locations for damage assessment, in: J. Abualdenien, A. Borrmann, L.-C. Ungureanu, T. Hartmann (Eds.), EG-ICE 2021 Proceedings: Workshop on Intelligent Computing in Engineering, Berlin, 2021, pp. 411–420, http://dx.doi.org/10.14279/depositonce-12021.

[39] A. Göbels, J. Beetz, Relative location ontology: An ontological model for representing directional topological relationships between spatial entities in oriented space, in: P. Pauwels, M.a. Poveda-Villalón, W. Terkaj (Eds.), Proceedings of the 12th Linked Data in Architecture and Construction Workshop, 2024, pp. 91–104, (Accessed 16 December 2024). URL: https://ceur-ws.org/Vol-3824/paper7.pdf.

[40] W.O.W. Group, Web ontology language (OWL), 2004, (Accessed 29 February 2024). URL: https://www.w3.org/OWL/.

[41] M. Grüninger, M.S. Fox, The role of competency questions in enterprise engineering, in: A.r. Rolstadås (Ed.), Benchmarking — Theory and Practice, Springer US, Boston, MA, 1995, pp. 22–31, http://dx.doi.org/10.1007/978-0-387-34847-6_3.

[42] N.F. Noy, D.L. McGuinness, Ontology Development 101: A Guide to Creating Your First Ontology, Technical Report, Stanford Knowledge Systems Laboratory, 2001, (Accessed 4 January 2024). URL: https://protege.stanford.edu/publications/ontology_development/ontology101.pdf.

[43] A. Borrmann, J. Blankenbach, J. Beetz, M. König, R. Becker, T.A.W. Blut, T. Celik, F. Benedikt, A. Göbels, M.S. Mafipour, J. Martens, M. Scheffer, S. Vilgertshofer, Technologien Zur Generierung Digitaler Zwillinge Als Grundlage Für Betrieb Und Instandhaltung Baulicher Infrastruktur, Technical Report, Bundesministerium für Digitales und Verkehr (BMDV), 2023, (Accessed 11 January 2025). URL: https://publications.rwth-aachen.de/record/973486.

[44] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, D. Novotny, VGGT: Visual geometry grounded transformer, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2025, pp. 5294–5306, http://dx.doi.org/10.1109/CVPR52734.2025.00499.

[45] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (6) (1981) 381–395, http://dx.doi.org/10.1145/358669.358692.

[46] H. Abdi, L.J. Williams, Principal component analysis, Wiley Interdiscip. Rev.: Comput. Stat. 2 (4) (2010) 433–459, http://dx.doi.org/10.1002/wics.101.

[47] M. Peng, A. Cui, C. Kang, S. Marx, AI-based extraction and management of text and view information from 2D bridge engineering drawings, in: A. Moreno-Rangel, B. Kumar (Eds.), Proceedings of the 32nd International Workshop on Intelligent Computing in Engineering, EG-ICE 2025, Glasgow, United Kingdom, ISBN: 9781914241826, 2025, http://dx.doi.org/10.17868/strath.00093312.

[48] S. Mafipour, D. Ahmed, S. Vilgertshofer, A. Borrmann, Digitalization of 2D bridge drawings using deep learning models, in: Proceedings of the 30th International Workshop on Intelligent Computing in Engineering, EG-ICE 2023, London, UK, 2023, pp. 611–618, (Accessed 10 February 2025).URL: https://www.ucl.ac.uk/bartlett/sites/bartlett/files/digitalization_of_2d_bridge_drawings_using_deep_learning_models.pdf.

[49] B. Faltin, P. Schönfelder, D. Gann, M. König, Reconstructing as-built beam bridge geometry from construction drawings using deep learning-based symbol pose estimation, Adv. Eng. Inform. 62 (2024) 102808, http://dx.doi.org/10.1016/j.aei.2024.102808.

[50] A. Göbels, J. Beetz, Graph-based method for extracting spatial information from semi-formal text to derive 3D bridge and damage models from legacy maintenance data, Autom. Constr. 178 (2025) 106418, http://dx.doi.org/10.1016/j.autcon.2025.106418.

[51] A.-H. Hamdan, M. Bonduel, R.J. Scherer, An ontological model for the representation of damage to constructions, in: M. Poveda-Villalón, P. Pauwels, R.D. Klerk, A. Roxin (Eds.), Proceedings of the 7th Linked Data in Architecture and Construction Workshop, Lisbon, Portugal, 2019, pp. 64–77, (Accessed 5 June 2024). URL: https://ceur-ws.org/Vol-2389/05paper.pdf.