

AN ONTOLOGY CONCEPT FOR THE TOPOLOGICAL ABSTRACTION OF INFRASTRUCTURE NETWORKS

Jens Bartnitzek¹, and Al-Hakam Hamdan¹

¹A+S Consult GmbH, Dresden, Germany

Abstract

Infrastructure planning requires the use of various geometric representations of traffic nets thus exchanging different data models during a project that all refer to the same domain object. In order to provide a common data basis for a BIM-based workflow, this article proposes a topological model for representing traffic nets based on the parametric planing objects. Therefore, a concept for connecting the topology with alignment geometry data is described. Furthermore, the proposed concept allows the abstraction of multiple macro topologies for various use cases that all refer to the same lower level micro topology. Additionally, the approach is implemented by utilizing Semantic Web Technologies and therefore developing an OWL ontology concept for representing traffic topologies. Subsequently the concept is exemplified through an ontology that defines the topology representation of a simple rail switch.

Introduction

Due to the inherent curvature of the earth of the engineering alignment and the large-scale expansions of rail networks, different types of georeferenced representations of the rail network are required. The alignment planning is based on projected geodetic coordinate reference systems such as Universal Mercator Projection (UTM) where the site planning is separated from the height planning. Consequently, a common base model is required to which the various geometric representations could refer. This is especially required when applying methods of Building Information Modeling (BIM) in infrastructure planning, whereby various stakeholders and domain-specific project participants are involved.

One possible approach would be the use of a model that represents only the topological relations of the infrastructure network. This way, different geometric representations could reference or be linked to the same consistent topology, thus preserving data integrity in digital planning processes.

An infrastructure topology allows disparate geometric datasets to integrate with a single logical network representation, avoiding inconsistencies. The topological connectivity also enables sophisticated network analysis for tasks like routing, vulnerability assessment, and optimization. Since the topology is decoupled from physical geometry, it supports flexibility in aligning to different coordinate projections or levels of detail.

A new concept for modeling a topology that represents traffic nets of any kind is proposed in this article. Theoretically, this concept could be implemented in a variety of data structures, e.g. as an SQL database or using BIM standards such as the Industry Foundation Classes (IFC).

Since topologies can be represented well as graphs and in order to integrate the underlying logic-based concepts in the model, the developed topology concept is implemented as a graph-based ontology using the Web Ontology Language (OWL). The proposed concept is demonstrated on an exemplary ontology that defines the topology representation of a simple rail switch.

Related Work

Various ontological approaches exist that aim for a topological representation of constructions. According to Gruber (2009), an ontology in the context of computer and information sciences is defined as a set of representational elements that can be used to model a domain of knowledge or discourse. These representational elements include classes, properties, and relationships or relations. Furthermore, the representational elements of an ontology contain information about their meaning as well as constraints for their logically consistent application. Ontologies can be created in various formats, of which the most prominent one is OWL that is also W3C-standardized¹. OWL is based on the Resource Description Framework (RDF)² and Resource Description Framework Schema (RDFS)³, which are also W3C standardized and allow the formulation of logical statements about any things (referred to as resources in RDF) in the form of triples consisting of subject, predicate and object. In this regard, OWL adds further concepts that enable the representation of knowledge on the basis of description logic (Baader et al., 2017). Since based on description logic, OWL-based ontologies are divided into two separate systems, one of which describes the Terminological Component (TBox) and the other the Assertional Component (ABox). In the TBox the knowledge concepts used in the ontology are defined including classes and properties. It establishes the hierarchy of classes and the specifications on properties, such as domain and range specifications, class equivalences, and property characteristics. The ABox contains assertions about instances of the classes defined in the TBox, essentially the factual knowledge. By reasoning the ABox against a TBox by an OWL-compatible reasoning engine, implicit information can be inferred based on the asserted facts in the ABox.

For buildings the Building Topology Ontology (BOT) (Rasmussen et al., 2021) can be used to define topological relations between digital representations of building zones and elements. Consequently, no geometry is involved in the topology, however references to geometry data can be

¹<https://www.w3.org/OWL/> (accessed: April 03, 2024)

²<https://www.w3.org/RDF/> (accessed: April 03, 2024)

³<https://www.w3.org/TR/rdf-schema/> (accessed: April 03, 2024)

made through links on BOT resources. A similar approach can be realized for bridges with the Bridge Topology Ontology (BROT) (Hamdan and Scherer, 2020). Both approaches have in common that the structure of a construction is realized through classes for zones and components. Comparatively, a traffic net is described on a more abstract level and wider scale consisting of stations and connecting tracks.

An ontology for representing this kind of structure has been developed by Lorenz et al. (2005) as Ontology of Transportation Networks (OTN). Thereby, not only stations and their respective connections are modeled but also corresponding land areas or built structures. Furthermore, the connections could be typed via classes, e.g. a connection can be classified as waterway, road or railway. Another ontology for representing the topology of rails has been developed as Rail Topology Ontology (RTO) by Bischof and Schenner (2021). RTO is an OWL representation of the RailTopoModel (RTM), which has been released as International Railway Standard 30100 and is defined as UML model. Compared to OTN, RTM and RTO allow the definition of different aggregation levels such as nano, micro or macro for modeling the topology in various detail levels, e.g. for visualizing the network in a broader scale. However, parallel definition of higher level topologies that refer to the same lower level topology are usually not supported by RTM or lead to inconsistencies. Furthermore, OTN as well as RTM and RTO define no alignment strategies in their respective topology concepts for connecting topological data objects exactly to related geometry background by using the geometric alignment stations. Besides ontological approaches, also general concepts for realising topologies in infrastructure have been developed. For instance, according to Bendfeldt (2005) a railway station can be divided:

- by operational technology in the areas of train transportation and train formation,
- operationally in the areas of route junctions and track groups or
- by type of transport in the areas of rail passenger transport and rail freight transport

Furthermore, Bendfeldt (2005) investigates how stations can be broken down into elementary nodes and elementary stations in order to derive standardization's and design principles for optimized operational and infrastructural aspects.

Methodology

In order to develop the topology and subsequently the prototypical ontology, core requirements must first be identified that have to be fulfilled by the resulting concepts. In this regard, a fundamental principle of traffic planning that needs to be considered for the topology is the need for specific traffic routes that result from the distribution of potentials. A location for a potential is defined as the position of an individual settlement or business location as well as an agglomeration of these locations. Topologically these

(combined) locations are abstracted as nodes. Traffic potentials exist between the nodes, which as directed edges describe the potential for a traffic relationship in the form of travel or goods transportation between the nodes. Lill's law of travel (see eq. 1) provides a simplified way of evaluating this potential (Lill, 1891).

$$V_{AB} := k \cdot \frac{P_A \cdot P_B}{d(A, B)^2} \quad (1)$$

The size of the possible traffic flow V_{AB} is proportional to the product of the size of the agglomerations P_A , P_B to be connected and inversely proportional to the square of their distances $d(A, B)$. Furthermore, k is an additional constant correction value (Lill, 1891).

The logical network of potentials is realized with a physical network of traffic infrastructure across several modes of transport. Track-guided traffic routes form an essential part of the physical network, but require a reference to the associated geometric representation. At the lowest topological level, the node-edge model represents the exact physical network, while other topological models generalize it through a topological abstraction. A topology can have a geometric representation, but it does not have to. Furthermore, the representation may differ from the physical realization. For instance, the geometric representation of a station can, for example, be realized geometrically via its station building, the arithmetic center of its points or an arbitrarily set station position.

Consequently, the following requirements for an infrastructure topology can be determined:

1. The topology should consist of multiple abstraction layers.
2. The lowest abstraction layer should represent the exact physical infrastructure network.
3. Higher abstraction layers could generalize the topological abstraction of the infrastructure network.
4. Since the lowest abstraction layer represents the exact physical infrastructure network, it should be possible that an optional geometrical representation could be mapped to the layer components.

The infrastructure topology can therefore have a geometric representation, but does not have to. This representation may differ from the physical realization. The geometric representation of a station can, for example, be realized geometrically via its station building, the arithmetic center of its switches or an arbitrarily set station position. This creates a strict separation of the levels:

- The route alignment is the level of physical realization of the network
- The topology is a logical level that can reference the route alignment in a specialization, but does not correspond to it.

After this initial step of identifying the topology requirements, a topology concept will be developed that aims to fulfil them (see section "Concept of the Infrastructure

Topology”). The concept should be applicable on various data structures. For an initial prototypical implementation of the resulting topology concept the research team has decided to develop an OWL ontology. This allows for representing the topologies as graphs which are flexible to modify, especially since new edges can be added to nodes without violating previously defined data schemes. Furthermore, OWL supports logic-based modeling allowing the integration of underlying knowledge concepts and their automatic reasoning. Additionally, by using OWL as machine-interpretable standard, the resulting topology concepts can be shared and connected with other knowledge domains. For developing the prototypical ontology the Linked open terms methodology (Poveda-Villalón et al., 2022) is used, which is structured in the four steps (1) ontology requirements specification, (2) ontology implementation, (3) ontology publication and (4) ontology maintenance. For proposing the initial concept in this article, only the first two steps are processed, followed by a publication of the prototype ontology as draft. The requirements for the ontology should be the same as for the overall topology concept (see requirements 1. - 4. above). During the ontology implementation it has been considered to reuse components of OTN and RTO (see section “Related Work”) directly in the newly developed ontology. However, since the structure of both ontologies differs greatly from the abstracted topology concept proposed in this paper, only the development of corresponding alignment ontologies is considered for future research.

Concept of the Infrastructure Topology

The proposed topology is an abstraction of an infrastructure network realized through logical nodes and edges that are separated from its geometric references. Therefore, the topology is defined as directed graph, which consists of the following elements:

- a set $V = \{v_0, \dots, v_n\}$ of nodes
- a set $E = \{e_0, \dots, e_m\}$ of directed edges between the nodes

Fig. 1 demonstrates an example of this abstract model consisting of 5 nodes and 8 edges that result in a directed graph.

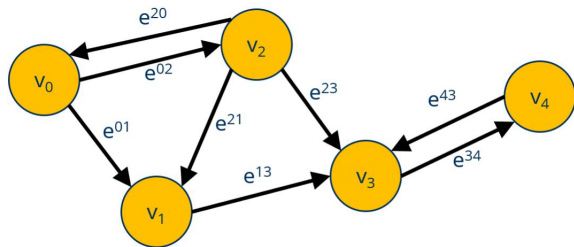


Figure 1: Directed graph consisting of 5 nodes and 8 edges

Topologically the edge $e_i := e^{jk} := (v_j, v_k)$ is unique between the nodes v_j and v_k . If there are more direct paths between v_j and v_k these are summarized topologically to e^{jk} . Through the graph the edges have access to their associated nodes and the nodes have access to their linked

edges.

The proposed topology utilizes microtopologies and macrotopologies as utilized levels for abstracting a route network.

Microtopologies

The microscopic topology (microtopology) of a transportation network describes the first topological abstraction of the network from the geometry to a node-edge model. It is also the only geometric abstraction. Fig. 2 presents an exemplary microtopology of a simple switch, in which the two possible paths shown between the nodes are physically defined via the main or branch track. However, an exact geometric location and demarcation between the switch and the rest of the track can only be realized by a topological representation for the start and end of the switch. The double crossing switch results in four possible paths, only two of which are described by the alignment of the two tracks. Thus the geometric descriptions of the two additional paths are therefore required as alignments and are later assigned to the switch object.

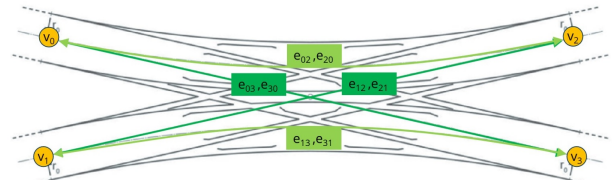


Figure 2: Exemplary microscopic topology of a single switch and a double crossing switch. The directions are indicated by double arrows. Background Figures from (Fendrich and Fengler, 2014)

If the topological decomposition of the switch is omitted, the geometric links to the physical network at these points are incomplete or only implicitly contained via the switch parameters. In addition, the switch is only positioned at a virtual intersection point, which is irrelevant for topological references back to the geometry.



Figure 3: Structure of the microtopology of a traffic junction of two roads (Fig. from Wollschläger (2021) overlaid with the microtopological nodes)

The abstraction corresponds to the definition of junctions in road engineering. As shown in Fig. 3, there is also a strict separation between the geometric construction and

the topological abstraction. The references to each other are established via nodes on the geometric lanes and edges as sections on the lanes.

Macrotopologies

The macroscopic topology (macrotopology) of a transportation network describes a topological abstraction of an existing topological network. It is therefore based on either a microscopic or macroscopic topology. It has only topological and no geometric references. Any number of macrotopologies can be constructed for a microtopology. An exemplary macrotopology is shown in Fig. 4.

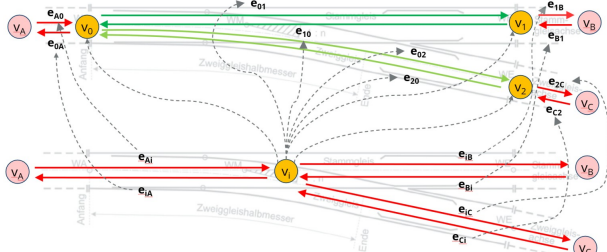


Figure 4: Abstraction of a switch as node v that aggregates nodes v_0 , v_1 , v_2

In the example a simple switch is defined through 3 inner nodes v_0 , v_1 and v_2 . According to the described data model, these nodes have back references to their linked edges, e.g. node v_0 to the edges e_{0A} , e_{01} and e_{02} with exact geometric back references.

When the microtopology is abstracted to a macrotopology, a macro node v is generated as a switch representation from the micro nodes v_0 , v_1 and v_2 . So that the macro node can form back references to the geometry without defining them itself, it must link all the associated micro nodes v_0 , v_1 and v_2 . For completeness, all inner edges e_{01} , e_{10} , e_{02} and e_{20} are also linked. Thus v itself is a directed graph.

The macro edges refer to their corresponding micro edges, e.g. e_{Ai} refers to e_{A0} , and if available their corresponding enclosed micro nodes.

All levels must be linkable with each other for a uniform, consistent topology approach. By linking them, arbitrarily abstract topology levels implicitly gain access to the underlying microtopology and thus to the physical geometry.

Structure of an OWL Topology

Based on the proposed concept of an infrastructure topology the Infrastructure Topology Ontology (INTO)⁴ has been formalized using OWL. INTO provides the necessary TBox to support the modeling of microtopologies and all macrotopologies at higher abstraction levels as well as additional linking components for connecting the topology with geometry. In this regard, micro- and macrotopologies are specializations of the abstract definition of a graph consisting of nodes and edges according to Fig. 1.

Table 1 contains an overview of the prefixes used for the realization of the ontology. The prefix *into* refers to a namespace that is used by all classes and properties specifically

⁴<https://app.korfin.de/ontology/into>

created for INTO. In this regard, the INTO ontology uses terminology from RDFS (prefix: *rdfs*) and OWL (prefix: *owl*).

Table 1: Namespaces and prefixes used in this article

Prefix	Namespace
rdfs	https://www.w3.org/2000/01/rdf-schema#
owl	https://www.w3.org/2002/07/owl#
into	https://app.korfin.de/ontology/into#

An important implementation aspect is that edges are not formalized as OWL object properties but instead as named individuals. The primary reason for this design decision is that not only nodes are referenced by lower and higher topology abstraction levels but also edges, thus requiring a form of linking mechanism for statements. One possible solution would be the use of the RDF extension RDF-star⁵, which allows the direct linking of complete RDF statements to other resources. However, RDF-star is not interpretable by most OWL reasoning engines, thus reducing the overall applicability of the ontology in an OWL context. The same applies to alternative graph models, e.g. the graph database Neo4J⁶. For this reason edges have been modelled as individuals according to the ontology design pattern "N-ary Relations"⁷. Thus, each instance of *into:edge* is connected to two instances of *into:node* via subproperties of *into:connectsToNode* but in addition can also be referenced by other instances of *into:TopologicalElement* of higher or lower topology layers. Direct connections in the form of auxiliary object properties are inferred by reasoning corresponding property chain axioms. The core concepts of INTO are shown in Fig. 5. There, each topology, be it a micro- or macrotopology, is represented as instance of *into:Topology*. Corresponding nodes (*into:Node*) and edges (*into:Edge*) are assigned to the topology through respective object properties (*into:containsNode* or *into:containsEdge*). In this regard, the *into:Topology* instance is used as container object for storing the topological components. The actual topology is defined by assigning *into:Node* instances as start and end-node to instances of *into:Edge* via the object properties *into:startsAtNode* and *into:endsAtNode*, thus implicitly defining the edge direction. The properties *into:startsAtNode* and *into:endsAtNode* are sub properties of *into:connectsToNode*, which is an inverse object property of *into:connectsToEdge*. Thus, by using a property chain axiom that infers the object property *into:isConnected* if an instance of *into:Node* is connected via *into:connectsToEdge* to an instance of *into:Edge* and that edge representation connects to a different instance of

⁵https://w3c.github.io/rdf-star/cg-spec/editors_draft.html (accessed April 03, 2024)

⁶<https://neo4j.com/> (accessed: April 03, 2024)

⁷<https://www.w3.org/TR/swbp-n-aryRelations/> (accessed: April 03, 2024)

into:Node via *into:connectsToNode* it is possible to model direct connections between nodes through reasoning.

The symmetric property *into:inverseToEdge* allows the definition of an edge that has an inverse direction compared to another edge and thus opposing start and end points.

Both classes *into:Node* as well as *into:Edge* are subclasses of the generalizing class *into:TopologicalElement*. Furthermore, *into:Node* and *into:Edge* have additional subclasses that are used for topological elements in macro-topologies (*into:MacroNode* and *into:MacroEdge*) and microtopologies that are linked with a geometry-based alignment (*into:AlignmentNode* and *into:AlignmentEdge*). An essential part of the proposed topological concept is the aggregation of topological elements, e.g. the aggregation of inner nodes in a macro node or macro edge. Therefore, each instance of *into:TopologicalElement* can aggregate a number of other instances of the same class via the object property *into:aggregatesTopologicalElement*. On the other hand the inverse relation via *into:isAggregatedByTopologicalElement* is also possible. Thus, inner nodes and edges as well as outer nodes and edges can be realized for each topological element.

OWL Realization of Microtopology

Modeling a microtopology is realized through alignment nodes (*into:AlignmentNode*) as specialized subtype of a node and alignment edges (*into:AlignmentEdge*) as specialized subtype of a common edge. These are the only types of nodes or edges, which are related to the alignment via alignment sections and thus are related to the rail geometry.

For each alignment section two alignment edges can be modeled, since in the directed, labelled graph each direction is represented through its own edge. Therefore, for each instance of *into:AlignmentEdge* a reference to the respective instance of *into:AlignmentSection* is necessary as well as a definition of the direction via the data property *into:direction*.

OWL Realization of Macrotopology

Macrotopologies are defined via macro nodes (*into:MacroNode*) as specialized subtype of nodes and macro edges (*into:MacroEdge*) as specialized subtype of edges. Macro nodes and macro edges connect respectively the corresponding nodes and edges of lower topology levels. For simplified calculations, the inner nodes are separated from the outer nodes and the inner edges from the outer edges in a macro node. A macro edge also has this separation, although it cannot have any outer nodes. Macro nodes and macro edges describe an independent, directed, labelled graph.

Thus infrastructure elements can be separated and abstracted from each other. The alignment is abstracted topologically via the microtopology including mapped switches. A topological track plan could be obtained by merging the switches into macro nodes and defining the

connecting macro edges. For example, if the macro nodes of all points of a station and possible multi-tracks between the stations are summarized in macro edges, it would be possible to get Line network plans. Furthermore, it is possible to determine comprehensive network maps by combining stations into agglomerations in further macro nodes as well as combining routes including "intermediate stations" into macro edges.

Connecting Topology with Geometry

To define the topological node-edge model independently from the underlying geometry, additional linking elements are used (see Fig. 7). The core elements for this are alignment sections (*into:AlignmentSection*) and alignment stations (*into:AlignmentStation*), which define the alignment positions. Instances of both of these classes can be assigned to an instance of *into:Alignment* that represents the alignment via the object properties *into:partAlignment* for linking alignment sections and *into:containsStation* for linking alignment stations. Additionally, an inverse object property of *into:partOfAlignment* named *into:dividedBySection* could be utilized in the modeling process.

An alignment section is defined through a start station and end station both positioned on the same alignment. Therefore, instances of *into:AlignmentStation* are assigned to an *into:AlignmentSection* instance via the properties *into:startsAtStation* and *into:endsAtStation*. Since the stations are defined as OWL individuals they can be reused in the modeling process, e.g. the end of an section is usually the start of the next section on the same alignment.

The configuration of an alignment section consisting of multiple sub-sections from different alignments, which could happen for example through different planning sections or sectional remeasurement, could be realized through a specialized alignment that is linking multiple alignment sections.

Thus the geometry reference is completely described and independent of the geometric construction, i.e. the topology can be defined at any point on the geometry.

The proposed concept can be generalized to point and line objects. Consequently, an instance of *into:AlignmentSection* represents a special line object and an instance of *into:AlignmentStation* a special point object. The further topological view is limited to the geometric reference, which is also defined by the generalization. Via property chain axioms alignment sections can also be assigned automatically to alignments if they define the start or end of a corresponding alignment section.

The connection between the alignment section and the topology is realized through linking instances of *into:AlignmentSection* and *into:Alignment* with micro edges realized through *into:Alignment*. Therefore, the object properties *into:assignedToSection* and *into:assignedToAlignment* are used. Furthermore, the relation *into:assignedToAlignment* between *into:AlignmentEdge* and *into:Alignment* can be inferred if the respective instance of *into:AlignmentEdge* is connected to an in-

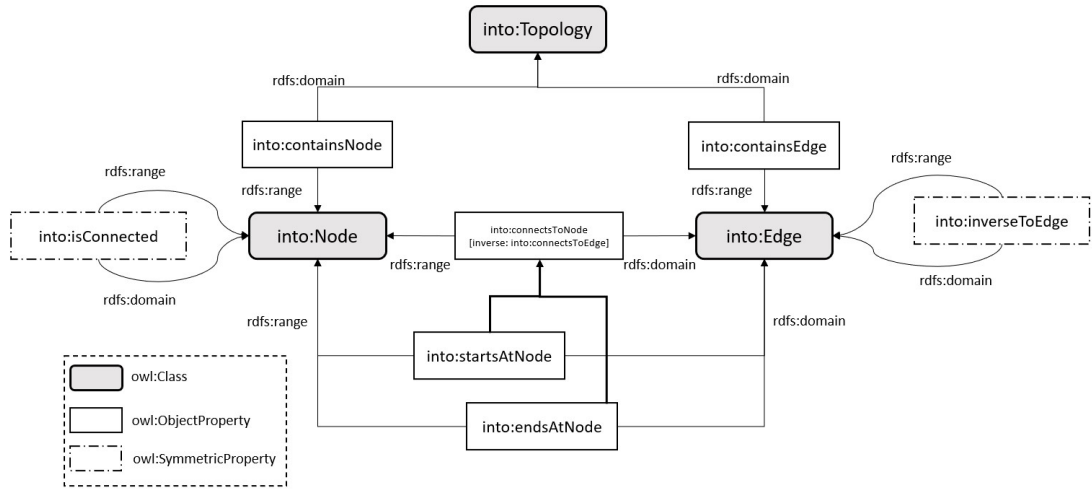


Figure 5: Core concepts of the topology part of the proposed OWL ontology

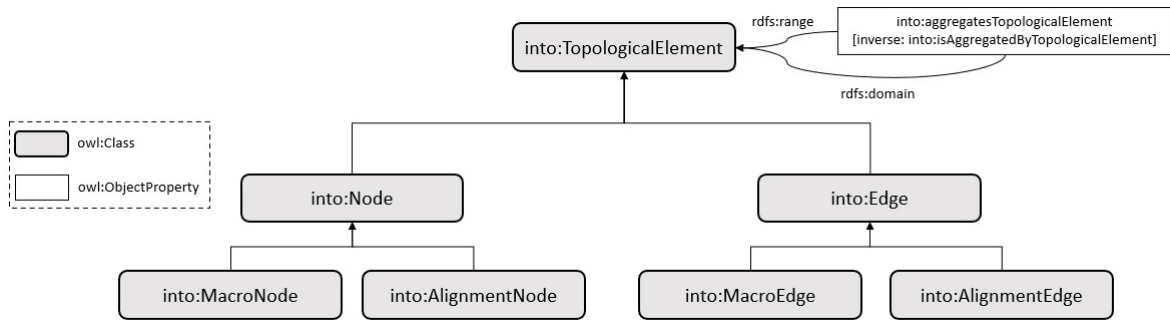


Figure 6: Inheritance concept for topological components

stance of *into:AlignmentSection* via *into:assignedToSection* and that representation of alignment section is linked to *into:Alignment* via *into:partOfAlignment*. The geometric data can either be linked directly in the OWL graph by using auxiliary ontologies such as the File Ontology for Geometry (FOG) (Bonduel et al., 2019) or by defining a separate link model, e.g. a linkset in an Information Container for Linked Document Delivery (ICDD) according to ISO 21597. Additionally, further management of the alignment representation could be utilized via the Ontology for Managing Geometry (OMG) (Wagner et al., 2019).

Application of the OWL Topology

To demonstrate the developed ontology an exemplary abstraction between two topological levels to represent a simple rail switch have been modeled as ABox using the TBox described in the previous section. The structure of the switch is the same as shown in Fig. 4. The topology example has been modeled as OWL ontology and uploaded in a locally hosted GraphDB⁸ triplestore. Fig. 8 shows a graph visualization whereby the focus node is v_i .

By reasoning the modeled assertional graph (ABox) that stores all factual information against the terminology (TBox) of the INTO Ontology, aggregations of topolog-

ical elements of the macro topology towards a microtopology can be inferred and vice versa. For instance, for the nodes v_0 , v_1 and v_2 the outer node v_i has been automatically inferred if v_i assigned these nodes as inner nodes. Furthermore, all inner edges e_{01} , e_{10} , e_{02} and e_{20} are also linked. Consequently, v itself is a directed, labelled graph. By processing the graph in a reasoning engine direct connections (*into:isConnected*) can be inferred for all nodes that belong to the same edge. For example, v_0 is connected to v_1 and v_2 , while v_1 and v_2 share no connection with each other.

For a high-performance evaluation of the node at runtime, it is advised to set up a path matrix across all outer nodes of the macro node, i.e. a statement whether v_1 can be reached via v_2 . However, this is not part of the ontology model.

The macro edges refer to their corresponding micro edges, e.g. e_{Ai} refers to e_{A0} , and if available their corresponding enclosed micro nodes.

⁸<https://graphdb.ontotext.com/> (accessed: April 03, 2024)

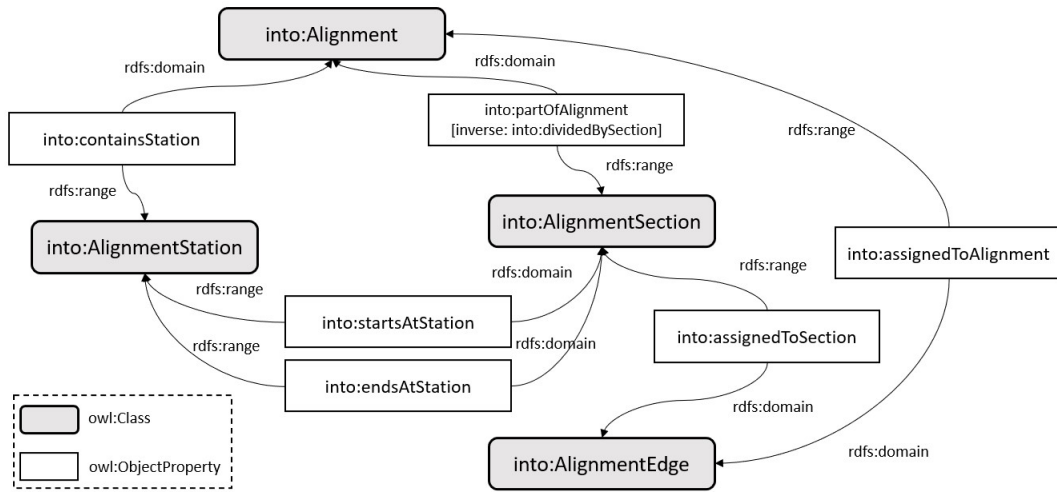


Figure 7: Concept for ontological alignment representation

The presented concept can also be used recursively for all higher-level macrotopologies.

In general the abstraction of the ontology involves the following steps:

1. Definition of the nodes and edges to be abstracted, so that the necessary nodes and edges can be clearly assigned to a higher abstraction level in a node-edge model.
2. Creation of the new macro nodes and internal structure by referencing the associated nodes and edges.
3. Creation of the new macro edges and of the inner structure by referencing the associated nodes and edges.
4. Linking the new macro nodes and macro edges via the back references of the outer nodes and edges.
5. Temporary creation of path matrices for macro nodes and macro edges via the inner structures.

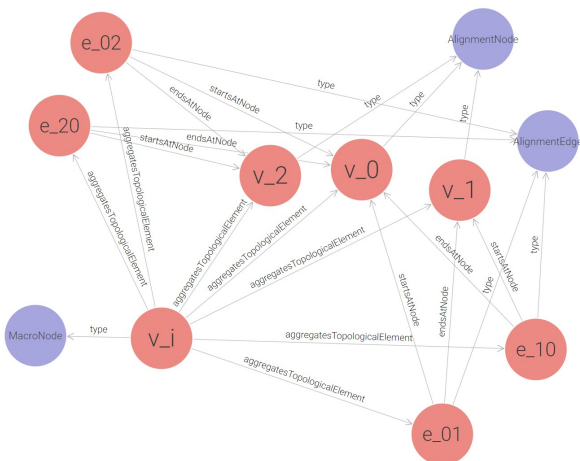


Figure 8: OWL ontology example in GraphDB

Based on the requirements defined in the "Methodology" section various SPARQL queries could be applied that

make use of the topology structure and inference mechanisms to provide information about the infrastructure network and validate the ontologies functionality. For instance Query 1 identifies elements of higher and lower abstraction layers for a set of topological elements.

Listing 1: SPARQL Query for Topological Aggregation

```

SELECT ?element ?lowerElement ?higherElement
WHERE {
  {
    ?element into:aggregatesTopologicalElement ?
      lowerElement .
    OPTIONAL {
      ?element into:isAggregatedByTopologicalElement ?
        higherElement .
    }
  }
  UNION
  {
    ?element into:isAggregatedByTopologicalElement ?
      lowerElement .
    OPTIONAL {
      ?element into:aggregatesTopologicalElement ?
        higherElement .
    }
  }
}

```

Based on the requirement that the lowest abstraction layer should represent the exact physical infrastructure network, queries for selecting information about instances of *into:AlignmentEdge*, e.g. in Query 2 could be applied. Thereby, Query 2 is used for determining the starting and ending station of each alignment edge. A similar query could be applied for instances of *into:AlignmentSection*.

Listing 2: SPARQL Query for Alignment Edges

```

SELECT ?edge ?start ?end
WHERE {
  ?edge a into:AlignmentEdge ;
    into:startsAtStation ?start ;
    into:endsAtStation ?end .
}

```

Conclusions

In this paper a concept for a topology that represents traffic nets has been proposed. In addition a corresponding prototypical ontology has been formalized in OWL and illustrated in an example of representing a simple switch.

The topology can be structured in multiple abstraction levels. In this regard two level types are used: (1) Micro-topologies that describe the traffic net at the lowest abstraction level and reference to a geometric representation and (2) Macrotopologies that describe a higher topological abstraction of the traffic net and therefore are based on lower microtopologies or macrotopologies. An important aspect is that multiple macrotopologies can refer to the same microtopology in parallel for various use cases, allowing the use of domain specific topologies for e.g. train transportation or route junctions.

One key aspect is that edges are not classified as OWL object properties when modeling the topological concept in an ontology. The main reason for this is that individuals that represent edges need to reference other edges in order to abstract the topology in multiple levels of different hierarchies. A combination of RDF-star and OWL has been considered but was not applied due to inference issues when processing the ontology in an OWL reasoning engine.

The topology can be used in various user scenarios. For example, a model for data exchange can be made available for traffic flow analyses and simulations. The topology can also act as the foundation of further BIM-supported infrastructure planning by providing a basic geometry-independent model. It is subject of future research how the developed ontology could be aligned with other ontologies used in infrastructure and construction planning, e.g. the Bridge Topology Ontology (BROT) (Hamdan and Scherer, 2020) or the Rail Topology Ontology (Bischof and Schenner, 2021). Furthermore, geodata ontologies such as the GeoSPARQL Ontology (OGC GeoSPARQL Standards Working Group, 2024) could be utilized for georeferencing the components of the topology. In order to ensure the practicability of the topology concept and the ontology, both developments must be applied in comprehensive test cases as part of future research.

References

- Baader, F., Horrocks, I., Lutz, C., and Sattler, U. (2017). *An Introduction to Description Logic*. Cambridge University Press.
- Bendfeldt, J.-P. (2005). *Möglichkeiten der Standardisierung in der Infrastrukturplanung von Eisenbahnknoten*. Eurailpress Tetzlaff-Hestra.
- Bischof, S. and Schenner, G. (2021). Rail topology ontology: A rail infrastructure base ontology. In *International Semantic Web Conference*, pages 597–612. Springer.
- Bonduel, M., Wagner, A., Pauwels, P., Vergauwen, M., and Klein, R. (2019). Including widespread geometry formats in semantic graphs using rdf literals. In *Proceedings of the 2019 European Conference on Computing in Construction*, volume 1 of *Computing in Construction*, pages 341–350, Chania, Greece. European Council on Computing in Construction. 10.35490/EC3.2019.166.
- Fendrich, L. and Fengler, W. (2014). *Handbuch Eisenbahninfrastruktur*. Springer-Verlag.
- Gruber, T. (2009). Ontology. In Liu, L. and Özsu, M. T., editors, *Encyclopedia of Database Systems*, pages 1963–1965, Boston, MA. Springer US. 10.1007/978-0-387-39940-9_318.
- Hamdan, A.-H. and Scherer, R. J. (2020). Integration of bim-related bridge information in an ontological knowledgebase. In *Proceedings of the 8th Linked Data in Architecture and Construction Workshop (LDAC)*, pages 77–90.
- Lill, E. (1891). *Das Reisegesetz und seine Anwendung auf den Eisenbahnverkehr mit verschiedenen auf die Betriebsergebnisse des Jahres 1889 bezugnehmenden statistischen Beilagen in Tabellen und bildlicher Form*. Spielhagen u. Schurich.
- Lorenz, B., Ohlbach, H. J., and Yang, L. (2005). *Ontology of transportation networks*. University of Munich, REWERSE Project, Tech. Rep. A1-D4.
- OGC GeoSPARQL Standards Working Group (2024). *Geosparql ontology*. <https://opengeospatial.github.io/ogc-geosparql/geosparql11/index.html>, accessed: 2024-01-30.
- Poveda-Villalón, M., Fernández-Izquierdo, A., Fernández-López, M., and García-Castro, R. (2022). Lot: An industrial oriented ontology engineering framework. *Engineering Applications of Artificial Intelligence*, 111:104755. <https://doi.org/10.1016/j.engappai.2022.104755>.
- Rasmussen, M. H., Lefrançois, M., Schneider, G. F., and Pauwels, P. (2021). Bot: The building topology ontology of the w3c linked building data group. *Semantic Web*, 12(1):143–161.
- Wagner, A., Bonduel, M., Pauwels, P., and Uwe, R. (2019). Relating geometry descriptions to its derivatives on the web. In *Proceedings of the 2019 European Conference on Computing in Construction*, volume 1 of *Computing in Construction*, pages 304–313, Chania, Greece. European Council on Computing in Construction. 10.35490/EC3.2019.146.
- Wollschläger, J. (2021). Implementierung von parametrischer geometrie, topologie und semantik für verkehrsknotenpunkte in georeferenzierten bim-projekten.