2026 European Conference on Computing in Construction
Corfu, Greece
July 12–15, 2026

ONTOLOGY-SUPPORTED BRIDGE COMPONENT DESIGN: IMPLEMENTATION OF BRIDGE CAP DESIGN RULES

Tim Noack[1], Han Qian[1], Chongjie Kang[1] and Steffen Marx[1]
[1]TUD Dresden University of Technology, Dresden, Germany

Abstract

The bridge design process is a complex and time-consuming task for bridge engineers. To reduce repetitive tasks, automating the application of guidelines and standards is desirable. Currently, automation attempts are mostly based on table-based calculations. This work proposes a new approach to automated, code-compliant structural design: an exemplary automation process for dimensioning the bridge cap. In this process, an ontology describing bridge components and boundary conditions is created. German road bridge design rules are implemented using the Semantic Web Rule Language (SWRL). The aim is to enable a fully automatic geometric dimensioning of bridge caps based on several boundary conditions.

Introduction

When designing a bridge, the bridge engineer encounters a challenging task. There is a vast number of boundary conditions, requirements and structural mechanics to consider. This leads to long planning and testing phases. Guidelines and standards have been created to support the engineers' decisions and to ensure the quality of bridge design. Their application is often repetitive and laborious because of the many required steps and decisions. Therefore, automating these tasks is desirable. In recent decades, this has primarily been achieved through table-based calculations and, in rare cases, by implementing parts of the standards in programming code (Jeske et al., 2022). Both approaches lack flexibility and are unsuitable for integration into an artificial intelligence-supported bridge design process, which will be essential for a future-proof and efficient bridge engineering sector.

Therefore, a new approach to automated, code-compliant structural design is proposed: an ontology-based bridge cap dimensioning process. Ontologies, as part of the Semantic Web, represent a way to structure semantic knowledge and make it machine-interpretable (Staab and Studer, 2009) and are therefore well suited for representing guidelines and standards. The dimensioning process of a bridge cap was chosen because it is highly standardized in German road guidelines. A bridge engineer needs only a limited set of boundary conditions to determine, without further decision-making, the appropriate bridge-cap type and its dimensions.

In the first step of this work, existing ontologies in the field of bridge engineering and construction guidelines are reviewed. Next, German bridge-engineering guidelines and the process of extracting design rules from them are described. Bridge-cap dimensioning rules are implemented in an ontology using the Semantic Web Rule Language (SWRL) together with existing and newly created ontology structures. The proposed bridge cap dimensioning process is carried out for an exemplary bridge. Finally, the integration of the approach into a Building Information Modeling (BIM)-based process is discussed.

State of the Art

Extensive research on ontologies has been conducted in the broader Architecture, Engineering and Construction (AEC) industry as well as in the more specialized field of bridge engineering. Farghaly et al. (2023) found that between 2000 and 2021, more than 140 journal articles on ontologies, the Semantic Web or Linked Data were published in the AEC sector. They were categorized into ten application fields, such as process, compliance, smart cities and sustainability. Most of the ontologies were developed for managing and interpreting infrastructure or building data. Others aim to create a semantic interoperability between different sets of information. The most established ontologies in this field are the IfcOWL ontology (Beetz et al., 2009) and the Building Topology Ontology (BOT) (Rasmussen et al., 2019, 2020). The purpose of both ontologies is to enable the mapping from Industry Foundation Classes (IFC) to an ontological representation. Thus, with these ontologies, building data from IFC files can be transformed into an ontological database. Since the IFC version 4.3, the standard also includes objects for infrastructure representation, specifically bridge elements and the new basic class IfcBridge (International Organization for Standardization (ISO), 2024). This is why an alignment of new bridge ontologies to the IFC standard or at least to IfcOWL or BOT is recommended.

The objective of this work is to create a new bridge component based on the bridge's boundary conditions and relevant guidelines and standards. Therefore, the ontology application field "compliance" is most important here and requires further investigation. Ontologies in this field have primarily been used for code-compliance evaluation, which is also addressed in this paper. Pauwels et al. (2011, 2017) describe the advantages of mapping building data from IFC-files (explicit building information) into a Semantic Web representation. By using rule languages such as SWRL or N3Logic together with an inference engine, implicit building information can be inferred. For example, the compliance of the building to a specific standard paragraph can be checked automatically. This was used in other research (Salama and El-Gohary, 2013; Jędrzejewska et al., 2026) but only for code-compliance checking, not for deriving a structural design directly from boundary conditions.

In the field of bridge engineering, a similar situation emerges. The application fields of developed ontologies are risk management (Hassan and Le, 2020), Structural Health Monitoring (Li et al., 2021), inspection (Erdene et al., 2020; Liu and El-Gohary, 2022; Zhang et al., 2022), diagnostics (Jäkel et al., 2025), bridge rehabilitation (Wu et al., 2021), and maintenance (Ren et al., 2019; Göbels and Beetz, 2025). These domains involve collecting, managing, and analyzing existing bridge data using various bridge ontologies. Hamdan and Scherer (2020) describe a foundational bridge ontology, the Bridge Topology Ontology (BROT), with several extensions. These ontologies can represent a bridge, its materials, components, and associated damages. BROT is based on BOT and therefore suitable for an integration into a BIM-based process.

Only few research studies have addressed ontologies in the field of bridge design. Aqlan et al. (2025) proposed an ontology for structuring data such as dimensions, structural details, boundary conditions, and images from bridge databases, making it accessible for machine-learning-based bridge design. Zhang et al. (2021) used an ontology for the holistic design of energy piles in a bridge deck deicing system. Although the ontology supports the design of these structures, the described principle is not an automatic design method based on the bridge's boundary conditions. Instead, it performs a database query to select which of 12 predefined design solutions best fits a given condition. Therefore, according to the author, no existing research has addressed automatic bridge-component design or dimensioning based on guideline-derived rules, despite the suitability of Semantic Web technologies for such tasks.

Methodology

Rules from Guidelines and Standards

Standards and guidelines contain information in four main formats: text, formulas, tables, and images (photos, drawings, diagrams). To digitalize and automate the document's content, the information must be interpreted and translated into *if-then* rules, formulas, or algorithms. The interpreted information can then be encoded and integrated into a digital process. The goal is to define boundary conditions as inputs and generate outputs —- such as parameter values or text —- after the process is executed. This process will be demonstrated in this work using a highly standardized bridge component: the bridge cap. Figure 1 illustrates the regular concrete road slab bridge cross section, including two bridge caps. The caps are a component that generally consists of reinforced concrete and carries the railings, traffic barriers and the foot- or cycleway.

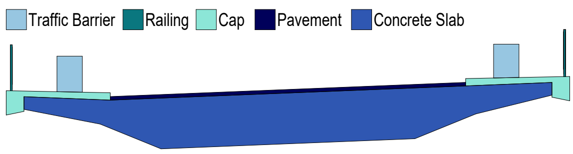

Figure 1: Superstructure cross-section: Main components of a road slab bridge.

In German standards and guidelines, there are several documents defining the design of concrete bridge caps. The RE-ING of the BASt (Bundesanstalt für Straßen- und Verkehrswesen, 2024) is a guideline describing the composition of road bridges. The RiZ-ING, also published by the BASt (Bundesanstalt für Straßen- und Verkehrswesen, 2023) is a collection of road bridge component drawings including twelve types of bridge caps. The Eurocode 2 (CEN/TC 250 Structural Eurocodes et al., 2023) and the DIN 1045 (DIN-Normenausschuss Bauwesen (NABau) et al., 2023) are specifying the concrete mixture and design details such as the concrete cover.

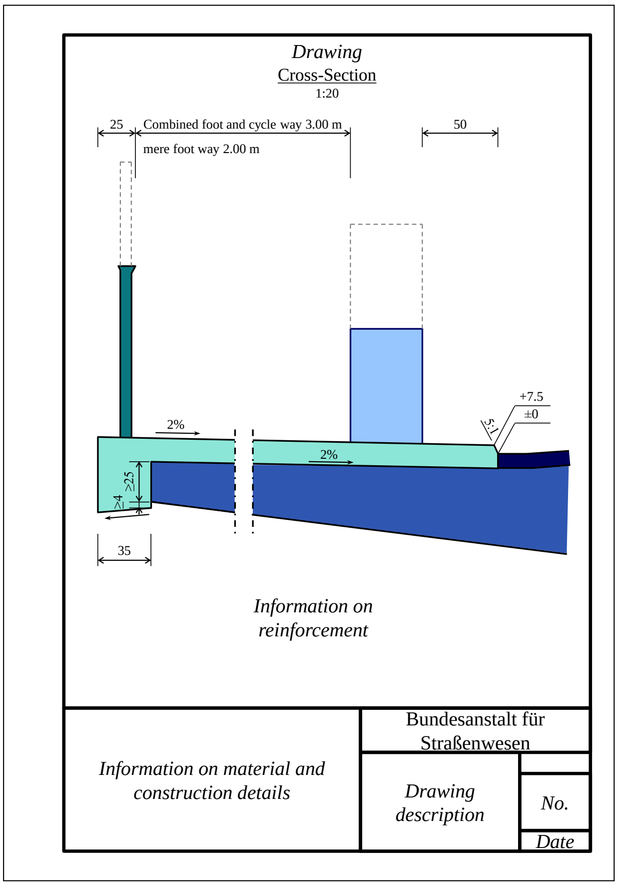

Figure 2: Composition of a RiZ-ING drawing with bridge cap measurements of drawing number "Kap 1 Blatt 3" (Bundesanstalt für Straßen- und Verkehrswesen, 2023)

This work primarily uses the RiZ-ING, since it focuses on early-stage design where material details and reinforcement configuration are not yet critical. The RiZ-ING drawings are generally A4-pages containing a drawing of the component and its adjacent components. In Figure 2, the composition of such a page is shown. Besides the drawing it contains indicated component dimensions and location as well as additional information in text-form in the drawing and the title block. Each drawing has a title describing the component and a number e.g., “Kap 1 Blatt 3” for the third page of the first cap drawing (Bundesanstalt für Straßen- und Verkehrswesen, 2023). From the information contained in these drawings, rules can be derived that describe the step-by-step dimensioning of bridge caps. These rules can then be used for automated bridge cap dimensioning. Because SWRL is used for implementation, all rules are expressed as *if-then* constructs. Some of these rules are described in the Tables 2, 3 and 4.

The information contained in the RiZ-ING cap drawings can be grouped into three categories:

A) Geometric information on the cap
B) Relational information between the cap and adjacent components
C) Additional information on the cap’s purpose, its material and its detailed design

To implement this information into an automated design process, relational and additional information is particularly critical. A minimum requirement would be implementing the rules in object-oriented programming (OOP) to correctly include all the information. OOP would be a good solution for the implementation of rigid *if-then* rules and the process in this work in general. However, OOP is not suitable when rules have to be implemented that are softer or contradicting and need to be weighted against each other. Here, an ontological structure would be appropriate with these rules described in the SPARQL Protocol And RDF Query Language (SPARQL) or the Shapes Constraint Language (SHACL) and applied using a Large Language Model (LLM). Therefore, this work uses ontologies to implement the automated bridge-cap dimensioning process and establish the foundation for future extensions.

Ontology

The ontological structure used in this work is based on the Bridge Topology Ontology (BROT) and the Bridge Component Ontology (BRCOMP) described by Hamdan and Scherer (2020), both of which build upon BOT introduced by Rasmussen et al. (2020). In addition, several *ObjectProperties* of the Relative Location Ontology (RELOC) by Göbels and Beetz (2024) have been incorporated.

Table 1 lists the ontologies used in this work. The Bridge Core Ontology (BCore) and the Bridge Element Design Rule Ontology (BEDRO) were created for this work in the Web Ontology Language (OWL 2) using Protégé (Musen, 2015). BCore extends BROT, BMAT and BRCOMP to represent the bridges, their components and their materials. BCore and BEDRO were created in a bottom-up approach. Thus, the *Classes* and properties

Table 1: Ontologies used in this work

Ontology	Prefix	IRI
Bridge Core Ontology	BCore	`https://w3id.org/bcore#`
Bridge Element Design Rule Ontology	BEDRO	`https://w3id.org/bedro#` *
Building Topology Ontology	BOT	`https://w3id.org/bot#`
Bridge Topology Ontology	BROT	`https://w3id.org/brot#`
Bridge Component Ontology	BRCOMP	`https://w3id.org/brcomp#`
Building Material Ontology	BMAT	`https://w3id.org/bmat#`
Relative Location Ontology	RELOC	`https://w3id.org/reloc#`

*The publication process for this website is not yet complete at the time of submission of this work.

necessary for the implementation of the RiZ-ING design rules were added to the basic framework of BROT, BMAT and BRCOMP. BCore includes structures for representing boundary conditions relevant to bridge design – the environment where the bridge is to be built. Additionally, it contains *Classes* and *ObjectProperties* describing agents (Persons or Companies) and media files (images, technical documents) that are connected to the specific bridge. These structures are not used in this paper, unlike the *Classes* or *ObjectProperties* displayed in Figure 3.

Here, the *Classes* needed for the dimensioning of the bridge caps are shown. A `Bridge` contains `Components`, such as the `EquipmentComponents` `Cap`, `TrafficBarrier` and `Pavement`. These components are also shown in Figure 1. Their relative locations are represented using the RELOC *ObjectProperties* `meetTop`, `meetBottom`, `meetLeft` and `meetRight`. This allows representing category B information. Newly created *Classes* are `InfrastructureRoute` and its subclasses `FootCycleRoute` and `RoadRoute`. The foot or cycle way can run to the left or to the right of a road. This information belongs to category C.

Several *DataProperties* were added to describe category A information, mainly geometric dimensions: the height of the pavement `PAVh`, the width of the traffic barrier `TRBb` and the cap’s geometric parameters shown in Figure 4. All these parameters have the assigned *Datatype* `xsd:decimal`. The `FootCycleWay` has a `xsd:boolean`

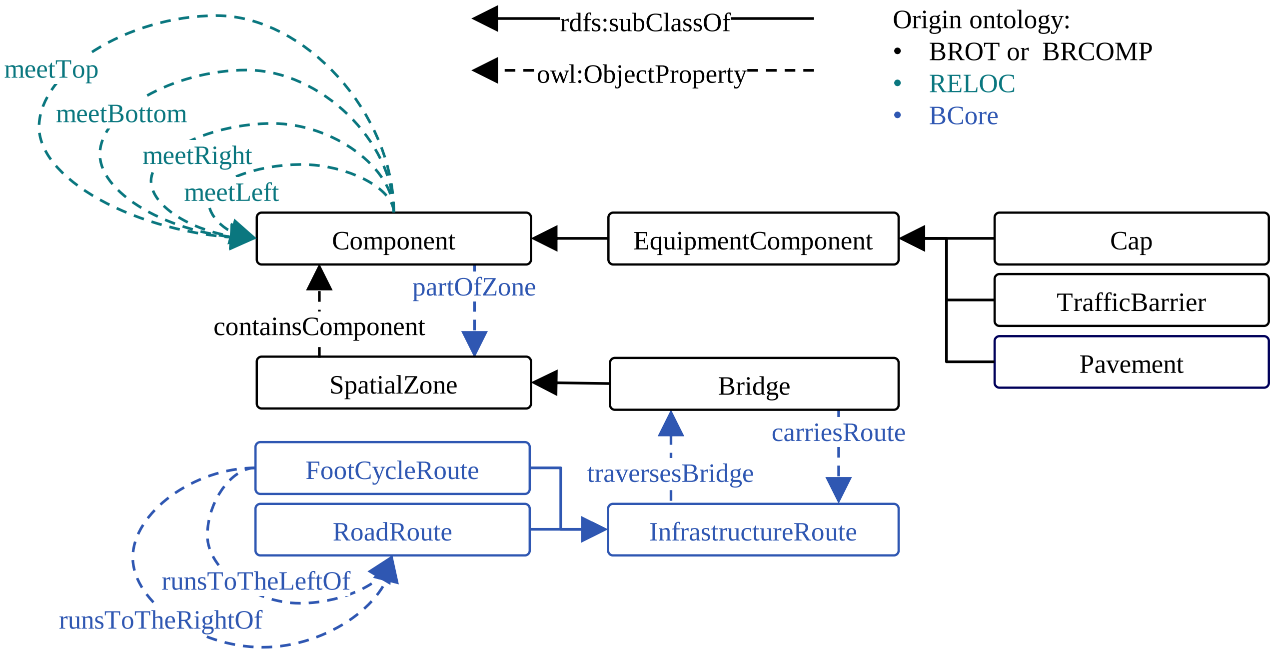

Figure 3: Diagram of Classes *from Hamdan and Scherer (2020) (BROT and BRCOMP, black), with added axioms of BCore (blue) and* ObjectProperties *from Göbels and Beetz (2024) (RELOC, green)*

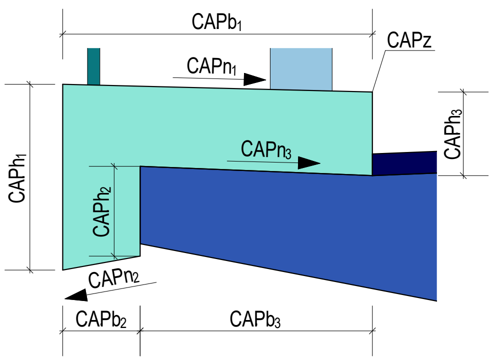

Figure 4: Geometric parameters in a bridge cap cross-section.

property `FootAndCycleWay`, which is `true` if it is a *combined* foot and cycle way. If it is only a foot way, it the value is `false`. The *DataProperty* `Position` of `Component` specifies, where the component is positioned in the bridge superstructure. The value 1 means left (relative to the direction of the route), the value 99 stands for right. If there are more than two components of the same type, they are numbered from left to right starting with 1, where right means always `Position = 99`. There are two more important *DataProperties* of `Cap`: the `CAPType`, indicating which cap design is assigned to the specific cap, and the `CAPCurb`, a `xsd:boolean`-Property describing if the cap has a curb or not. For example, a cap requires a curb if no traffic barrier is located above it, and vice versa.

This rule and other rules have been implemented in an ontology named BEDRO, an extension of BCore containing *if-then* rules extracted from the RiZ-ING guideline (Bundesanstalt für Straßen- und Verkehrswesen, 2023). The rules were first written in natural language before being translated into SWRL (O'Connor et al., 2005).

Three of these rules are contained in the tables below. They depict the cap dimensioning process. First, the boundary conditions of the bridge and the cap have to be checked to decide which type of cap to chose (Table 2). The type of cap refers to the different RiZ-ING drawings. For example, the drawing in Figure 2 dictates the measurements of an outer cap carrying a foot or cycle way and a traffic barrier. Inner caps are just needed if the bridge consists of multiple partial structures.

Table 2: Rule cap type*: description and SWRL-implementation*

If a road traverses a bridge and a combined foot and cycle way runs to the right of it and on the right side of the bridge it contains a cap with a traffic barrier on top, the cap's type is an outer cap with foot and cycle way and traffic barrier ("OuterCap_FCW_TRB")

```
RoadRoute(?r) ^FootCycleRoute(?fcw)
^runsToTheRightOf(?fcw,?r)
^FootAndCycleWay(?fcw,true)
^Bridge(?b) ^carriesRoute(?b,?r)
^carriesRoute(?b,?fcw)
^containsComponent(?b,?c) ^Cap(?c)
^Position(?c,99) ^meetBottom(?c,?trb)
^TrafficBarrier(?trb)
-> CAPType(?c,"OuterCap_FCW_TRB")
```

In the second step, the values contained in the RiZ-ING drawing are assigned to the *DataProperties* of the specific cap (Tables 3 and 4). They differ from cap type to cap type and are sometimes dependent on the dimensions of the adjacent components (`Pavement` and `TrafficBarrier`).

Table 3: Rule "Kap 1 Blatt 3" with combined foot- and cycle way*: description and SWRL-implementation*

If a cap has the type "OuterCap_FCW_TRB" (Outer cap with combined foot- and cycle way and traffic barrier) and it lies directly below a traffic barrier, the cap does not need a curb, its dimensions are declared as shown in Figure 2 and the top width is calculated as follows $CAPb_1 = 0.25 + 3.00 + TRBb + 0.5$ (unit: m)

```
Cap(?c) ^CAPType(?c,"OuterCap_FCW_TRB")
^meetBottom(?c,?trb) ^TrafficBarrier(?trb)
^TRBb(?trb,?trbb)
^swrlb:add(?b01,0.25,3.00,?trbb,0.50)
-> CAPb1(?c,?b01) ^CAPb2(?c,0.35)
^CAPn1(?c,0.02) ^CAPn3(?c,0.02)
^CAPh2(?c,0.29) ^CAPCurb(?c,false)
```

Table 4: Rule cap without curb*: description and SWRL-implementation*

If a bridge contains a pavement and a cap without curb, the cap's height 3 ($CAPh_3$) is 7.5 cm bigger than the pavement's height ($PAVh$), as shown in Figure 2

```
Bridge(?b) ^containsComponent(?b,?p)
^Pavement(?p) ^PAVh(?p,?ph)
^containsComponent(?b,?c) ^Cap(?c)
^CAPCurb(?c,false)
^swrlb:add(?ch3,?ph,0.075)
-> CAPh3(?c,?ch3)
```

The third step is to calculate the rest of the cap's dimensions (see Figure 4) using the following equations (for better readability, the prefix "CAP" has been removed):

$$n_2 \approx 7.5\,\% \tag{1}$$

$$b_3 = b_1 - b_2 \tag{2}$$

$$h_1 = h_2 + h_3 + b_1 n_1 + b_2 n_2 - b_3 n_3 \tag{3}$$

These equations can also be implemented using SWRL and its math built-ins. As soon as all the rules are implemented and are not contradicting, the ontology can be used for an automated bridge cap dimensioning process.

Automated Design Process

Bridge design is typically an iterative process involving repeated refinement of the bridge structure and the components. An exemplary process is shown in Figure 5. It begins with the assessment of the bridge's boundary conditions, followed by a rough design of the superstructure, substructure, and equipment. Then, it is checked if the boundary conditions are met and wether the design meets the requirements related to cost, environmental impact, and aesthetics.

This work proposes automating one part of this process – the equipment planning, specifically the dimensioning of bridge caps. The created ontology with the implemented cap design rules, BEDRO, can be used as follows:

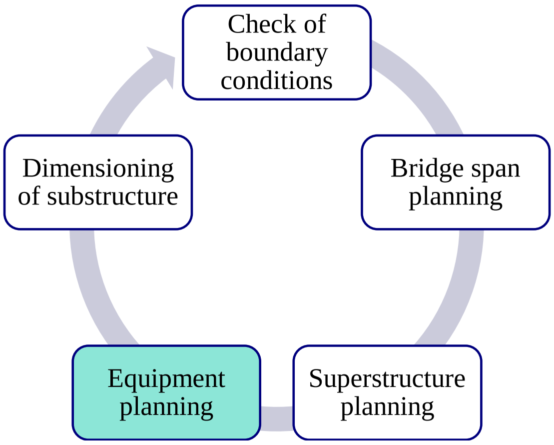

Figure 5: Iterative bridge design process.

1. Create instances of `Bridge`, `RoadRoute` and possibly one or two instances of `FootCycleRoute`.
2. Connect `RoadRoute` and `FootCycleRoute` instances to the `Bridge` via `traversesBridge` and assign either `runsToTheRightOf` or `runsToTheLeftOf` as between the routes.
3. Create an instance of `Pavement`, two instances of `Cap` and possibly instances of `TrafficBarrier`.
4. Assign *DataProperties* values for the components. Each cap and traffic barrier needs a value for its `Position`. The left component is assigned a "1", the right one a "99". As no design rules for `Pavement` or `TrafficBarrier` are implemented yet, their dimensions (e.g., `PAVh`, `TRBb`) must be provided manually.
5. Instantiate `partOfZone` as property of the components and the `Bridge`.
6. Run a SWRL-capable reasoner on the ontology and the dimensions of the bridge cap get automatically assigned values. In Protégé, the reasoner *Pellet* is recommended, as it handles SWRL effectively (O'Connor et al., 2005).

In Table 5, exemplary instances of a bridge and its corresponding routes and equipment components are described as well as their *ObjectProperties* and *DataProperties*. After implementing them in BEDRO and running the *Pellet*-reasoner, new relations and attributes have been created. Figure 6 shows the instance "B1Cap1", which is the left cap of the bridge "B1". The reasoner has inferred two *ObjectProperties*, describing that the cap is directly below the traffic barrier "B1TrB1" and directly left of the bridges pavement "B1Pav1". Also, it has inferred that the type of the cap is "OuterCap_FCW_TRB", following the rule in Table 2. The other inferred values are assigned by executing the other rules described above. "B1Cap99" does not have the same type as "B1Cap1" and hence has other values for its properties.

Since not many bridge engineers are experienced in working with ontologies or Protégé, the process must be made more user-friendly for practical adoption. A graphical user-interface (GUI) may be implemented,

Table 5: Exemplary instances and assigned properties for an automated dimensioning of the bridge caps "B1Cap1" and "B2Cap99"

Class	Instance	ObjectProperty/ DataProperty	Value
`Bridge`	B1		
`Road`	R1	`traversesBridge`	B1
`FootCycleWay`	FC1	`FootAndCycleWay`	true
		`runsToTheLeftOf`	R1
		`traversesBridge`	B1
`Cap`	B1Cap1	`Position`	1
		`partOfZone`	B1
`Cap`	B1Cap99	`Position`	99
		`partOfZone`	B1
`Pavement`	B1Pav	`PAVh`	0.07
		`partOfZone`	B1
`TrafficBarrier`	B1TrB1	`Position`	1
		`TRBb`	0.5
		`partOfZone`	B1

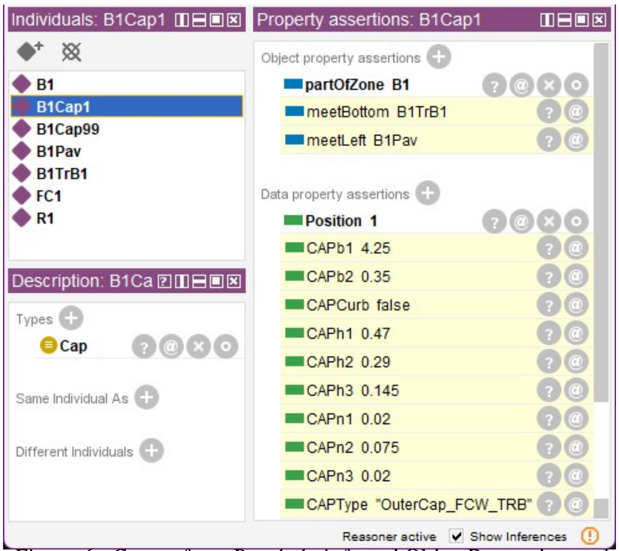

Figure 6: Cutout from Protégé: inferred ObjectProperties *and* DataProperties *of "B1Cap1"*

where the engineer can insert the boundary conditions of the respective bridge. Via an application programming interface (API), the *Classes* and *Properties* are then created, the reasoner is run and the generated cap dimensions are displayed as in Figure 7. Also, an export of these dimensions to a CSV-file would be feasible.

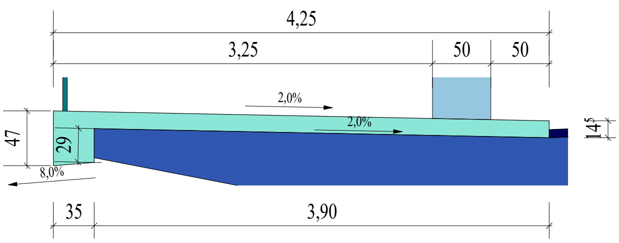

Figure 7: Rule-conform bridge cap cross-section.

A step further would be to implement the whole process into a combined GIS-BIM-approach. Here, it would be possible to extract the relevant boundary conditions from geographical information system (GIS)-data. The boundary conditions are then integrated into the ontology via an API. After the application of the rules using a reasoner, the generated component attributes can be extracted. With this data, a parametric building information model as described by Jeske et al. (2022) is populated, to obtain rule-conform dimensions and material information. This model can be used for further analyses such as quantity or cost estimation, as well as for visualization of the created bridge components.

Discussion

This work proposes a method for implementing design rules from a German guideline, the RiZ-ING, into an ontology. This ontology can be used to automatically generate rule-conform bridge cap dimensions for specific bridge boundary conditions. For that, SWRL rules were derived and can be applied using a reasoner. This process is limited in its user friendliness, applicability and scalability.

As stated before, bridge engineers do not usually work with ontologies and even when they do, many manual steps are still required before the cap dimensions can be generated. A solution could be a GUI, or integration into a GIS-BIM-application. Implementing a GUI or API would be simpler if the entire process were programmed in OOP. No API to the ontology would be needed and *if-then* rules can also be implemented in OOP. SWRL also has limitations: although it can in principle create new instances or properties, most reasoners do not support this functionality. Also, the Open World Assumption of OWL 2 makes it hard to check for non-existent instances (O'Connor et al., 2005).

The integrated rules apply only to new German road bridge caps. Thus, when designing a railway bridge or a French bridge, the ontology cannot be used. Also, the rules from the RiZ-ING the RiZ-ING rules are recommendations rather than legal requirements i.e., it is possible to deviate from these rules. Fixed *if-then* rules do not allow such deviations which also complicates transferring the approach to other components or another structure. In this case, a highly standardized component is needed, where every dimension can be derived using rules. In the field of bridge design, this condition mainly applies to equipment components e.g., the pavement, railing or traffic barrier. But, possibly, a manual intermediate step is needed here for a complete component design. For an automated design of the whole bridge, generative artificial intelligence is needed, but can be supported by rules implemented in an ontology.

Nevertheless, this exemplary automated dimensioning shows the potential of ontology applications in the construction sector. As stated above, in the AEC sector, ontologies and semantic rules have not yet been used to create new structural designs from the structure's boundary conditions. Especially in the field of bridge engineering, using ontologies for bridge design and guideline implementation is a new topic.

In principle, it is possible to transfer the process to other bridge components if they are as highly standardized as the bridge cap. The implementation of different kinds of rules still needs to be tested. Also, a human in the loop-implementation is imaginable if not every design decision can be automatized. When embedded in a well-adapted application, it could save bridge engineers significant time by avoiding repeatedly dimensioning the bridge cap or other standardized components every time for a new round of design circle (see Figure 5) but just to enter the boundary conditions and to generate the dimensions automatically. With that, they can concentrate on the more important decisions in the substructure and superstructure planning.

Conclusions

Bridge design consists of a vast amount of boundary conditions to consider and of decisions to make. This work proposes a way of automating the dimensioning of highly standardized bridge components, to reduce the engineer's workload. German road bridge guidelines are translated into *if-then*-rules defining the dimensions of the bridge caps for specific boundary conditions. The rules are implemented in an ontology using SWRL. The ontology in combination with a reasoner can be used to automatically generate the cap's dimensions. The process was applied to an exemplary imaginary bridge. The findings of this work are:

1. Information in guidelines and standards is contained in different forms and formats. For an automated design process, the information needs to be translated into *if-then*-rules, which can be transformed into programming code and applied to the respective design task. In this work, the *if-then*-rules were transformed into SWRL-rules.
2. It is possible to generate the bridge cap's dimensions using an ontology, SWRL and a reasoner. For that, the boundary conditions of the bridge and the bridge components need to be created as instances in the ontology. The proposed process can be transferred to other components, as long as they are highly standardized and the content of these standards can be translated into *if-then*-rules.
3. There are several limitations of this process: user friendliness, applicability and expandability. These limitations can be met using different state of the art-technologies such as a GIS-BIM-application, generative AI and newly developed APIs or by creating a well-adapted GUI and manual intermediate steps and intermediate decisions into the bridge design process.

Acknowledgments

The research project "KI-basierter Brückengenerator mit Wissens- und Erfahrungsdaten und früher Bürgerbeteiligung - HyBridGen" and the research project "Nutzung digitaler Regelwerke zur effizienten Planung und Prüfung von BIM-basierten Brückenentwürfen für die Eisenbahn - SmartBridgeRules" are funded by the German Federal Ministry for Transport within the framework of the mFUND funding program (Grant No. 01FV2067A and 01F1228B).

References

Aqlan, S., Hamdan, A.-H., Marx, S., and Kang, C. (2025). Ontology-driven Database for Integrating Bridge Data from Multiple Open Sources.

Beetz, J., Van Leeuwen, J., and De Vries, B. (2009). IfcOWL: A case of transforming EXPRESS schemas into ontologies. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 23(1):89–101.

Bundesanstalt für Straßen- und Verkehrswesen (2023). Richtzeichnungen für Ingenieurbauten - RiZ-ING.

Bundesanstalt für Straßen- und Verkehrswesen (2024). Richtlinien für den Entwurf, die konstruktive Ausbildung und Ausstattung von Ingenieurbauten - RE-ING.

CEN/TC 250 Structural Eurocodes, CEN/TC 250/SC 2 Eurocode 2: Design of concrete structures, and NA 005-07-01 AA Design of concrete structures (national mirror committee for CEN/TC 250/SC 2) (2023). EN 1992 - Eurocode 2 - Design of concrete structures.

DIN-Normenausschuss Bauwesen (NABau), DIN Standards Committee Building and Civil Energeering, and NA 005-07-01 AA Design of concrete structures (national mirror committee for CEN/TC 250/SC 2) (2023). DIN 1045 - Concrete Structures.

Erdene, K., Kwon, T. H., and Lee, S.-H. (2020). Integration of Extended IFC-BIM and Ontology for Information Management of Bridge Inspection. Journal of the

Computational Structural Engineering Institute of Korea, 33(6):411–417.

Farghaly, K., Soman, R. K., and Zhou, S. A. (2023). The evolution of ontology in AEC: A two-decade synthesis, application domains, and future directions. Journal of Industrial Information Integration, 36:100519.

Göbels, A. and Beetz, J. (2024). Relative Location Ontology: An ontological model for representing directional topological relationships between spatial entities in oriented space. In Proceedings of the 12th Linked Data in Architecture and Construction Workshop, pages 91–104, Bochum, Germany. CEUR-WS.

Göbels, A. and Beetz, J. (2025). Graph-based method for extracting spatial information from semi-formal text to derive 3D bridge and damage models from legacy maintenance data. Automation in Construction, 178:106418.

Hamdan, A.-H. and Scherer, R. J. (2020). Integration of BIM-related bridge information in an ontological knowledgebase. In 8th Linked Data in Architecture and Construction Workshop - Proceedings, Online.

Hassan, F. U. and Le, T. (2020). Ontology-Based Decoding of Risks Encoded in the Prescriptive Requirements in Bridge Design Codes. Kitakyushu, Japan.

International Organization for Standardization (ISO) (2024). ISO 16739 - Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries.

Jeske, A., Scheil, M., and Castrischer, U. (2022). Parametrisierte Brückenmodellierung mit PythonParts in Allplan. Bautechnik, 99(7):565–574.

Jäkel, J.-I., Heinlein, E., Morgenstern, H., Kim, H., and Klemt-Albert, K. (2025). System- and Data-integrated linking of digital 3D models of existing bridge structures with Knowledge Graphs of non-destructive diagnostic methods. Journal of Information Technology in Construction, 30:603–630.

Jędrzejewska, A., Leonardi, M. L., Ramonell Cazador, C., Granja, J., Chacón, R., Oliveira, D. V., and Azenha, M. (2026). Eurocode Core Ontology: Building the semantic framework for structural design assistants. Advanced Engineering Informatics, 70:104218.

Li, R., Mo, T., Yang, J., Jiang, S., Li, T., and Liu, Y. (2021). Ontologies-Based Domain Knowledge Modeling and Heterogeneous Sensor Data Integration for Bridge Health Monitoring Systems. IEEE Transactions on Industrial Informatics, 17(1):321–332.

Liu, K. and El-Gohary, N. (2022). Bridge Deterioration Knowledge Ontology for Supporting Bridge Document Analytics. Journal of Construction Engineering and Management, 148(6):04022030.

Musen, M. A. (2015). The protégé project: a look back and a look forward. AI Matters, 1(4):4–12.

O'Connor, M., Knublauch, H., Tu, S., Grosof, B., Dean, M., Grosso, W., and Musen, M. (2005). Supporting Rule System Interoperability on the Semantic Web with SWRL. In The Semantic Web – ISWC 2005, volume 3729, pages 974–986. Springer Berlin Heidelberg, Berlin, Heidelberg. Series Title: Lecture Notes in Computer Science.

Pauwels, P., De Farias, T. M., Zhang, C., Roxin, A., Beetz, J., De Roo, J., and Nicolle, C. (2017). A performance benchmark over semantic rule checking approaches in construction industry. Advanced Engineering Informatics, 33:68–88.

Pauwels, P., Van Deursen, D., Verstraeten, R., De Roo, J., De Meyer, R., Van De Walle, R., and Van Campenhout, J. (2011). A semantic rule checking environment for building performance checking. Automation in Construction, 20(5):506–518.

Rasmussen, M. H., Lefrançois, M., Pauwels, P., Hviid, C. A., and Karlshøj, J. (2019). Managing interrelated project information in AEC Knowledge Graphs. Automation in Construction, 108:102956.

Rasmussen, M. H., Schneider, G. F., and Pauwels, P. (2020). BOT: the Building Topology Ontology of the W3C Linked Building Data Group. Semantic Web Journal.

Ren, G., Ding, R., and Li, H. (2019). Building an ontological knowledgebase for bridge maintenance. Advances in Engineering Software, 130:24–40.

Salama, D. A. and El-Gohary, N. M. (2013). Automated Compliance Checking of Construction Operation Plans Using a Deontology for the Construction Domain. Journal of Computing in Civil Engineering, 27(6):681–698.

Staab, S. and Studer, R., editors (2009). Handbook on ontologies. International handbooks on information systems. Springer, Berlin, 2 edition.

Wu, C., Wu, P., Wang, J., Jiang, R., Chen, M., and Wang, X. (2021). Ontological knowledge base for concrete bridge rehabilitation project management. Automation in Construction, 121:103428.

Zhang, P., Cui, C., Li, C., Zhang, C., and Liu, H. (2021). Holistic Design of Energy Pile Bridge Deicing System With Ontology-Based Multiobjective Decision Making. Frontiers in Materials, 8:710404.

Zhang, Y., Liu, Y., Lei, G., Liu, S., and Liang, P. (2022). An Enhanced Information Retrieval Method Based on Ontology for Bridge Inspection. Applied Sciences, 12(20):10599.