SHACLEval – A Quality Framework for the Shapes Constraint Language*

Achim Reiz[1,4,*,†], Henrik Dibowski[2,†], Irlan Grangel Gonzales[2,†], Fajar J. Ekaputra[3,†] and Kurt Sandkuhl[4]

[1] *Neonto GmbH, Weyertal 109, 50931 Cologne, Germany*
[2] *Robert Bosch GmbH, Bosch Research, Reni, Germany*
[3] *Vienna University of Economics and Business (WU), Vienna, Austria*
[4] *Rostock University, Rostock, Germany*

[†] These authors contributed equally.

Abstract

Semantic Web technologies have transformed the processing and representation of data. Initially used for linking publicly available knowledge, it is now widely adopted in enterprise contexts. Enterprise knowledge graphs (KGs) often use the Shape Constraint Language (SHACL) to validate data structure and completeness. SHACL constraints validate whether newly ingested data conforms to business and data rules, ensuring that data conforms to self-set standards and is interoperable in the long term. However, these constraints can be complex and demanding to manage, as they continue to develop to cater with the variety and complexity of the data they validate. Therefore, it is crucial to ensure the quality of such restrictions. One way of measuring the quality of SHACL shapes is through *ontology metrics* that translate the qualitative nature of ontologies into objective quantitative measurements. Over the past few years, various ontology metric frameworks have been published. However, they are often targeted for inference languages like OWL and fail to address the validation specifics of SHACL. This paper fills this gap by presenting SHACLEval, an evaluation framework for SHACL. SHACLEval proposes measures that assess the specific SHACL-language constructs. The novel metrics link the data strategy with relevant KPIs, enabling the detection of potential discrepancies between the KG strategy and development execution. The case is motivated by a Bosch Use-Case and demonstrated on a public SHACL repository.

Keywords

SHACL, Ontology Quality, NEOntometrics, Data Quality

1. Introduction

With their ability to connect heterogeneous knowledge, Knowledge Graphs (KGs) are at the forefront of transforming data silos into shareable knowledge. In the past, they have been primarily driven by academia. However, the increasing variety and velocity of enterprise data motivates an increasing use in the industry as well [1].

However, the open-world assumption of the prevalent ontology standards OWL or RDFS was deemed counterintuitive for people primarily interested in data modeling. While they allow the definition of complex inference rules, they do not allow the modeling of business rules to be entered into schema-like data models that define structure. This lack led to the creation of SHACL with a focus on data validation [2].

Regardless of the tasks, the ontologies must be of high quality to deliver value. There is already an extensive body of literature regarding the quantitative assessment of ontologies, primarily focusing on inference tasks. They consider, for example, graph attributes, annotations, or inheritance

* *First International Workshop on Users and Knowledge Graphs (UKG), September 3rd, 2025, Vienna, Austria*
[1,*] Corresponding author.
[†] These authors contributed equally.
achim.reiz@neonto.de (A. Reiz); henrik.dibowski@de.bosch.com (H. Dibowski); irlan.grangelgonzalez@de.bosch.com (I. Gonzalez-Grangel); fajar.ekaputra@wu.ac.at (F. Ekaputra); kurt.sandkuhl@uni-rostock.de (K. Sandkuhl);
0000-0003-1446-9670 (A. Reiz); 0000-0002-9672-2387 (H. Dibowski); 0000-0001-8460-4132 (I G. Gonzalez); 0000-0003-4569-2496 (F. J. Ekaputra); 0000-0002-7431-8412 (K. Sandkuhl)

patterns. However, the SHACL validation perspective brings some specific challenges beyond today's frameworks' capabilities.

To name a few, RDFS and OWL structure properties and classes along an inheritance hierarchy. Every instance of a class deeper in the hierarchy is also a member of the higher-level class. SHACL, in contrast, does not describe inheritances but is meant to be used in conjunction with RDFS and OWL. Depending on the engines' capabilities and setup, the shapes can be attached to classes using an RDFS or OWL entailment regime. However, shapes can also target single individuals or instances that are the subject or object of a given property. Also, SHACL has further capabilities for restricting potential values on literals, e.g., using REGEX or value ranges, and allows setting cardinality restrictions on properties.

The specifics of SHACL are beyond the scope of today's frameworks. Nevertheless, with the growing adoption of the language, there is a need to evaluate the developed constraints to ensure their quality and fitness for use. In this paper, we target the gap by proposing SHACLEval, an evaluation framework for SHACL constraints. SHACLEval proposes 25 measures for assessing the inner fabrics of these constraints. Using these measurements, enterprises can connect their data strategy for KGs to meaningful Key Performance Indicators (KPIs). This connection ensures that self-set goals are met, leading to higher KG quality and subsequent applications.

The rest of the paper is structured as follows. In Section 2, we recapitulate the related work on ontology evaluation. Afterward, we introduce the SHACLEval framework in Section 3, with the symbols and the underlying measurements, followed by an overview of how to derive meaningful KPIs from the measurements in Section 4. The practical application is motivated by two Bosch Use-Cases and an exemplary analysis of a public repository in Section 5 and 6 respectively. Finally, we conclude and outline future work in Section 7.

2. Related Work

In this section, a review and critical discussion of the current approaches related to SHACL evaluations is carried out. Raad and Cruz [3] categorize the existing evaluation efforts into four categories: Gold Standard, Application/Task-Based, Data-Driven, and Criteria-Based. The first approach compares the current ontology to a perfect one. Application/Task-based measures how well an ontology performs in each context. Data-driven uses a (e.g., textual) corpus to assess the ontology coverage. Criteria-based methods describe methods that evaluate the fit of an ontology to desirable structural or metaphysical attributes.

In this categorization, SHACLEval is a criteria-based structural assessment. It uses the number of occurrences of certain SHACL constructs to derive conclusions on its development. The rest of the section introduces further structure-based evaluation frameworks and the evaluation specifics of SHACL.

2.1. Existing Ontology Evaluation Frameworks

While there has been little activity regarding SHACL-specific evaluations, evaluating computational ontologies is a more mature research field. Various research methods have been proposed to assess the graph structure or OWL-specific vocabulary.

Over time, various surveys gathered state-of-the-art information. In 2016, Porn et al. published a systematic literature review on OWL evaluation approaches. The authors extracted eleven ontology quality criteria and assessment techniques and then used these criteria to categorize the papers [4]. In 2021, Wilson et al. reviewed existing quality criteria and measurements and categorized them into five categories: syntactic, structural, semantic, pragmatic, and social [5]. This review is until spring 2024, the most recent review on ontology quality metrics.

Tartir et al. proposed the OntoQA framework [6,7]. OntoQA proposed metrics based on the classes, their relationships, inheritances, instantiations, and connected attributes. A unique feature of this framework is the definition of measurements not only for the ontology as a whole but also for actual classes and relations.

Gangemi et al. built the oQual O² ontology evaluation design pattern [8,9]. Part of it is various measurements that assess the graph structure (here: structural dimension) and the inheritance path structures. Besides the structural evaluations, the authors describe the functional and usability profiling dimension, similar to the metric categorization in [3].

The OQuaRE framework by Duque-Ramos et al. translated the ISO 25000/SQuaRE software quality measurement methodology for ontologies [10]. The authors propose measurements and desirable metric values and associate these measurements with quality characteristics. However, further independent studies identified heterogeneity in the framework and raised doubts about the claimed statements and the framework's real-world applicability [11].

Fernández-Izquierdo et al. proposed a framework that not only regards the structural attributes of an ontology but also attributes from the design phase, coming from an ontology requirement specification document or ontology requirement testing suite [12]. Examples of these metrics are the number of requirements or test cases. While the authors regard tests as SPARQL-based evaluations, it could be argued that SHACL can also perform these SPARQL-based validations. Thus, their framework is a potential connection point to the SHACL evaluation, as it can be implemented using SHACL shapes.

Several authors were concerned with the cohesion and modularity of an ontology. Yao et al. developed a cohesion measurement framework, assessing the interconnections within an ontology [13]. Ma et al. proposed cohesion measurements focusing on detecting inconsistencies [14]. Oh et al. built an ontology module evaluation based on software evaluation research [15].

Outside of ontology evaluation, the Semantic Web community is also looking to improve the quality of various other semantic artefacts. The quality assessment of Linked Data / Knowledge Graphs is one of them, where Zaveri et al. [16] proposed a set of 18 quality dimensions alongside 69 metrics linked to these dimensions. These dimensions are further categorized into four different categories: Accessibility, Representation, Contextual, and Intrinsic.

Another work is focusing on assessing the quality of R2RML mappings [17] where they focus on four different metrics to assess the quality of mappings, including (a) usage of undefined classes, (b) usage of undefined properties, (c) usage of blank nodes, and (d) mapping quality reports. They extended the Luzzu Framework, initially developed for Linked Data quality assessment, to conduct the assessment [18].

2.2. SHACL-Evaluation

SHACL became a W3C standard in 2017 [19]. That makes it, comparatively to OWL and RDFS, a recent development. Current research on SHACL is mainly concerned with decision problems and semantics on recursive constraint declarations and corresponding validation implementations [20].

There was only limited activity regarding the evaluation of the shapes themselves. As part of the SHACTOR shape extraction tool, Rabbani et al. provide basic quantitative measures for node and property shapes [21]. However, these metrics are not intended for a general SHACL evaluation but to fine-tune the shape retrieval attributes.

Lieber et al. collected the statistics on the usage of SHACL axioms in publicly available data on GitHub. The authors intended to find SHACL constructs that are not commonly used and argue that these shapes need further attention in the corresponding modeling software [22]. However, there is currently no quantitative evaluation framework available.

Some approaches aim to generate SHACL from existing knowledge graphs. For example, Spahiu et al. created a knowledge base profiler that aims at creating SHACL heuristically from instance data [23]. ASTREA is an endeavor to automatically create SHACL shapes based on an existing ontology. As part of their evaluation process, the authors quantitatively assess the automatically created shapes of eight existing knowledge graphs [24].

2.3. Research Gaps

Today's ontology measurement frameworks primarily focus on graph traits and generalizable characteristics, like annotations, classes, or attributes. Some frameworks, like [6,7,25] also regard instances (thus going beyond mere ontology). That allows these frameworks to cover a wide range of potential ontology languages, making them a good fit for hierarchical graphs, like RDF(S) based ontologies or the various OWL profiles.

However, SHACL has some specifics that are not covered by a graph or class-based evaluation perspective. The data validation of SHACL is focused on constraints for cardinalities, attribute values, and potential paths. SHACL shapes are not instantiated but are applied to individuals directly or to classes, which are instantiated by rdf:type axioms. This lack of measures covering the structural validation attributes motivated the creation of the SHACL-specific SHACLEval framework.

3. The SHACLEval Framework

Motivated by the need to evaluate the developed constraints to ensure their quality and fitness for use, we proposed the SHACLEval framework. The framework assesses various characteristics of SHACL constraints using pre-determined metrics. In total, we identified 25 different metric elements. It quantifies, among others, the use of various constraint types and their application to classes or individuals. It also evaluates how the shapes integrate with existing ontologies and data. We will provide more details about these elements in Section 3.1.

Built upon these elements, we developed a set of evaluation metrics for SHACL constraints, which can be used directly for evaluating SHACL constraints in specific use cases and contexts. These metrics are categorized into several categories, such as type constraints (e.g., the total number of comparative property-pair constraints) and property constraint metrics (e.g., the ratio of how many of all cardinality constraints are min-cardinality constraints). Details about these constraints are provided in Section 3.2.

To use the proposed metrics in the real-world context, we proposed to adapt an existing process: *Requirements-Oriented Methodology for Evaluation Ontologies* (ROMEO) [26], consisting of three steps: (a) identification of evaluation requirements, (b) development of evaluation questions, and (c) identification of relevant SHACLEval metrics for each evaluation question. The details of this adaptation are provided in Section 3.3.

3.1. The SHACLEval Measurement Elements

SHACLEval proposes measures that evaluate the usage of SHACL vocabulary in a given graph. The framework has two levels: At the core of the SHACLEval framework are measurements of the W3C-defined, standardized SHACL, OWL, and RDFS vocabularies. The evaluation considers the ontology with the SHACL-graph and the data graph containing the instances. The second level is the use and combination of these underlying metrics in the framework presented in the following section.

Table 1 displays the elements that are being considered by the framework and represent quantitative measures. The capital letters represent the unrestricted number of elements, e.g., S, Cls. Function declarations $f(x)$ are used to describe restrictions of f on x. E.g., $Ns(Cls)$ stands for the number of classes restricted by node shapes. Finally, subscripts $X_{condition}$ indicates a condition. Here, only the shape X that has the attribute $condition$ are evaluated. E.g., $Ns_{wNonVal}$ describes SHACL-NodeShapes that have non-validation elements, like sh:message or sh:group.

The SHACL standard allows some degree of flexibility regarding entailment and reasoning behavior. While it is possible to require a specific entailment regime with `sh:entailment`, declaring a regime is optional. Thus, the actual validation results differ in correspondence to the validation software, and the proposed measures in Section 3.2 shall be interpreted considering the entailment regime used by the validation engine and the underlying validation use case.

Table 1
Measured Elements in the SHACLEval Framework.

Vocabulary	Symbol	Meaning
SHACL	S	The sum of explicitly declared shapes (node and property shapes).
SHACL	Ns	The sum of explicitly declared (not nested) node-shapes.
SHACL	Ps	The sum of explicitly declared (not nested) property-shapes.
SHACL	$X_{wNonVal}$	The sum of shapes with non-validational elements: sh:severity, sh:message, sh:name, sh:description, sh:order, sh:group, sh:defaultValue.
RDFS, OWL	Cls	The number of defined classes. E.g., implicitly by rdfs:subClassOf statements or explicitly by owl:Class statements.
SHACL	$Ns(Cls)$	The sum of all constrained classes.
SHACL	$Ns_{direct}(Cls)$	The sum of all classes that are only directly constrained (thus, not via sub-class relationship) by a node shape. Also possible with $NS_{indirect}$ and $NS_{direct\&indirect}$.
RDF, OWL	Ind	The sum of owl:Individuals.
SHACL	$S(Ind)$	The sum of individuals that are constrained by SHACL shapes.
SHACL	$S_{\in S(Ind)}$	The sum of shapes that constrain individuals.
SHACL	Ps_{min}	The sum of property shapes with only a minimum cardinality. There also exists Ps_{max} for only maximum cardinality constraints and $Ps_{min\&max}$ for both.
SHACL	Ps_{minVal}	The sum of property shapes with a minimum value constrained. There also exists Ps_{maxVal} for maximum value constraints.
OWL	Dp	The total number of datatype properties.
SHACL	$Ps(Dp)$	The number of datatype properties constrained by SHACL-shapes.
OWL	Op	The total number of object properties.
SHACL	$Ps(Op)$	The number of object properties constrained by SHACL-shapes.
SHACL	Pp	The number of pairwise property constraints: sh:disjoint, sh:equals, sh:lessThan, sh:lessThanOrEquals.
SHACL	$Ps_{valType}$	The number of constraints that restrict the value of a property: sh:class, sh:datatypes, sh:nodeKind.
SHACL	$Ps_{valRange}$	The number of constraints that restrict the value range of a numerical value: sh:minInclusive, sh:minExclusive, sh:maxInclusive, sh:maxExclusive.
SHACL	Ps_{string}	The sum of constraints that limit the potential value of an attached textual (string) value: sh:minLength, sh:maxLenth, sh:pattern, sh:uniqueLang.

The SHACL standard allows some degree of flexibility regarding entailment and reasoning behavior. While it is possible to require a specific entailment regime with `sh:entailment`, declaring a regime is optional. Thus, the actual validation results differ in correspondence to the validation software, and the proposed measures in Section 3.2 shall be interpreted considering the entailment regime used by the validation engine and the underlying validation use case.

The measurements of Table 1, which serve as the base for the SHACLEval framework, can be collected with SPARQL queries. We have developed the SPARQL queries to retrieve these basic measurements, which are available in our evaluation repository[2].

3.2. SHACLEval Evaluation Metrics

The measurements of elements in Table 1 is the basis for composing the SHACLEval framework. Building on these, we propose a set of 25 metrics, grouped in five categories that assess various constraint types, e.g., for literals; (object and data) properties; usage ratio on individuals or classes; and the existence of non-validational SHACL axioms.

Literal Constraints. The first evaluation metrics category assesses the **constraints** on **Literals** (cf. Table 2), measuring the restrictions for typed individuals. Thus, this category measures the number of restrictions that limit the use of typing (`a` or `rdf:type`) statements.

The measures allow for identifying how many data types are encoded in the shapes and whether the number of constraint types gets larger or smaller. *For example,* the data strategy might set the goal that the numerical attributes have value ranges corresponding to the domain, *e.g., age minimum and maximum.* The fulfillment level can be tracked using the measure $Ps_{valRange}$.

Table 2
SHACLEval – Literal Constraints

Calculation	Meaning
$Ps_{valType}$	The total number of value type constraints
$Ps_{valRange}$	The total number of value range constraints
Ps_{string}	The total number of string constraints
Pp	The total number of comparative property pair constraints.

Property Constraints. The property constraints metrics in Table 3 measure restrictions on data and object properties. They assess the number and kind of cardinality constraints (thus, how many relationships or attributes are allowed of a given type), how many of all properties are constrained, and how many of the constraint properties are literals (Data Properties Dp) or relationships to other elements (Object Properties Op).

The metrics give an overview of whether the SHACL validations cover the property elements of the ontology. Thus, these metrics are at the core of translating business to data rules and can identify potential imbalances between the existing data and the rules. *For example,* the data strategy aims to constrain all existing data properties with SHACL constraints. However, the $Ps(Dp)ToDp$ measure decreases, indicating a potential mismatch between data strategy and execution.

Table 3
SHACLEval – Property Constraint Metrics

Calculation	Meaning
$Ps_{minmax}Total = Ps_{min} + Ps_{max} + Ps_{min\&max}$	The total number of cardinality constraints.
$Ps_{min}Ratio = \frac{Ps_{min} + Ps_{min\&max}}{Ps_{min} + Ps_{max} + Ps_{min\&max}}$	The ratio of how much of all cardinality constraints are min-cardinality constraints.
$Ps_{max}Ratio = \frac{Ps_{max} + Ps_{min\&max}}{Ps_{min} + Ps_{max} + Ps_{min\&max}}$	The ratio of how much of all cardinality constraints are max-cardinality constraints.
$Ps_{valConst} = Ps_{minVal} + Ps_{maxVal}$	The total number of value constraints.

[2] https://doi.org/10.5281/zenodo.16538766

$Ps(Op)Ratio = \frac{Ps(Op)}{Ps(Op) + Ps(Dp)}$	The ratio of how much of the total constrained properties are object properties
$Ps(Dp)Ratio = \frac{Ps(Dp)}{Ps(Op) + Ps(Dp)}$	The ratio of how much of the total constrained properties are data properties
$Ps(Op)ToOp = \frac{Ps(Op)}{Op}$	The ratio of how much of the total object properties are constrained.
$Ps(Dp)ToDp = \frac{Ps(Dp)}{Dp}$	The ratio of how much of the total data properties are constrained.
$Ns(Cls)$	The total number of constrained classes.

Class Constraints. The class constraints metrics of Table **4** identify the number of class definitions constrained by SHACL shapes. The measures are distinct between classes targeted by shapes directly (e.g., by sh:class axioms) or indirectly (through sh:class in combination with RDFS entailments).

The metrics indicate how precisely the classes are constrained. *For example,* for top-level shapes, it might be desirable to target a high $Ns_{indirect}(cls)Ratio$, indicating that the top-level constraints are efficiently propagated down to the domain-specific classes. Furthermore, one may also create highly domain-specific shapes not meant to be used outside a given scope. This measurement is highly dependent on the validation engine and its entailment regime.

Table 4.
SHACLEval – Class Constraints

Calculation	Meaning
$Ns(Cls)ToCls = \frac{Ns(Cls)}{Cls}$	The ratio of how many of the total classes are constrained by SHACL shapes.
$Ns_{direct}(Cls)Ratio = \frac{Ns_{direct}(Cls)}{Ns(Cls)}$	The ratio of how many of all constrained classes are only constrained directly, without inheritance.
$Ns_{indirect}(Cls)Ratio = \frac{Ns_{indirect}(Cls)}{Ns(Cls)}$	The ratio of how many of all constrained classes are only constrained indirectly through inheritance.
$Ns_{direct\ \&\ indirect}(Cls)Ratio = \frac{Ns_{direct\&indirect}(Cls)}{Ns(Cls)}$	The ratio of how many classes are constrained both directly or indirectly through inheritance.
$Ns(Cls)$	The total number of constrained classes.

Individual Constraints. The individual constraints metrics (Table 5) measure how many individuals are restricted by SHACL shapes and the average number of restricted individuals per shape. Thus, the measures assess the connection between the schema (TBox) and the data (ABox).

They indicate how granular the SHACL shapes constrain the data and how much of the instance data is constrained by shapes. *For example,* if an organization has a goal that possibly all individuals have corresponding data rules, the fulfillment of this goal can be traced using the $S(Ind)ToInd$ metric.

Table 5.
SHACLEval – Individual Constraints

Calculation	Meaning
$S(Ind)ToInd = \frac{S(Ind)}{Ind}$	The ratio of how many of the total individuals are constrained by SHACL shapes.
$ShapeEffiency = \frac{S(Ind)}{S_{\in S(Ind)}}$	The average of individuals of each shape that constrain individuals.

Non-validation. These measures (cf. Table 6) assess the usage of human-targeted descriptions, like custom error messages, or information to build forms, like grouping or ordering.

Table 6.
SHACLEval – Non-Validational Elements

Calculation	Meaning
$Ns_{wNonVal}$	The sum of node shapes with non-validational elements.
$Ps_{wNonVal}$	The sum of property shapes with non-validational elements.
$Ns_{wNonVal}Ratio = \frac{Ns_{wNonVal}}{Ns_{wNonVal} + Ps_{wNonVal}}$	The node shapes in relation to all shapes with non-validational elements.
$Ps_{wNonVal}Ratio = \frac{Ps_{wNonVal}}{Ns_{wNonVal} + Ps_{wNonVal}}$	The property shapes in relation to all shapes with non-validational elements.
$Ns_{wNonVal}ToNs = \frac{Ns_{wNonVal}}{Ns}$	The ratio of how many of all node shapes have non-validational information attached.
$Ps_{wNonVal}ToPs = \frac{Ps_{wNonVal}}{Ps}$	The ratio of how many of all node shapes have non-validational information attached.

While validation is at the core of the SHACL standard, the validation results should be meaningful to developers, e.g., through human-readable descriptions. *For example,* an organization might set the goal that every node shape should be described with human-targeted information. Then, the $Ns_{wNonVal}ToNs$ should be close to 1.

3.3. Using Metrics for Quality Control

While metrics allow for empirically and objectively assessing the created SHACL shapes, they alone do not guarantee a practical evaluation. These measures must be carefully selected and interpreted to use them as meaningful KPIs. At the core of the interpretation is an alignment of the data strategy with appropriate measurement instruments (thus, metrics).

To achieve this alignment, Yu et al. proposed the "*Requirements oriented methodology for evaluating ontologies (ROMEO)*" [26]. Romeo is a top-down instrument used to find relevant evaluation KPIs. It builds on Basili et al.'s Goal Question Metric (GQM) approach [27] and makes some extensions that address the specifics of ontology engineering, mainly by providing templates for data gathering.

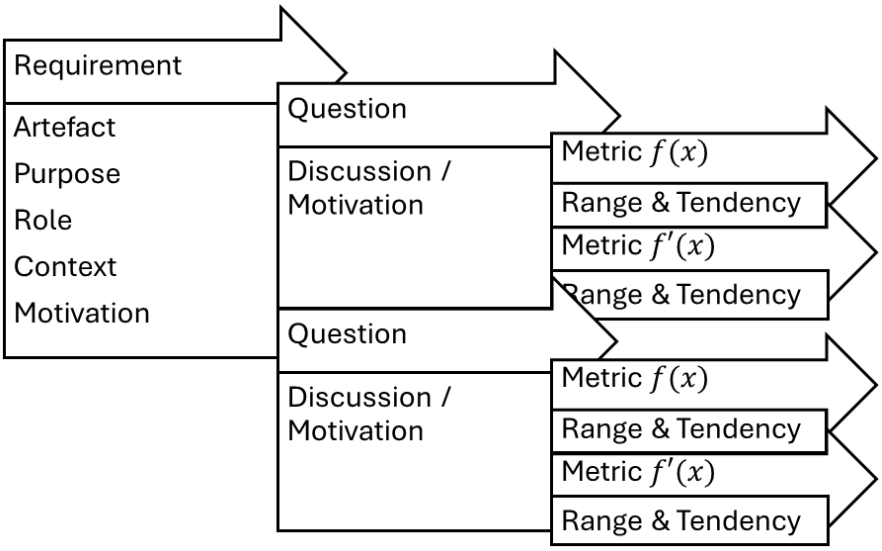

Figure 1. The process of requirement elicitation in the ROMEO framework.

In the ROMEO method, depicted in Figure 1, the first step is to identify structural requirements from existing functional ontology requirements, corresponding application requirements, or based on a newly performed requirement analysis. Afterward, each requirement is assigned one or more questions, which are then aligned with measurements. In the templates, every decision is explicitly argued and discussed.

ROMEO enables us to reach an agreement on what constitutes good quality between the various stakeholders. Its templates guide a KPI selection process that considers the various roles of the development process.

Table 7 - Table 9 provides a fictive example for assessing top-level SHACL shapes. For the sake of simplicity, we laid out only one requirement and connected it to one example question, which was assessed using two metrics. However, real-world evaluation scenarios like those depicted in the section 4 will have more complex assessment documentation.

Table 7.
ROMEO: Example Requirement Specification for SHACLEval.

Requirement SH_Top1: Adherence of data to top-level ontologies	
Analyze Artifact:	Reference Data Rules
For the purpose of:	Evaluating adherence to data standards
With respect to:	Sufficient Usage in Domain Ontologies to facilitate reusability
From the viewpoint of:	*Data Manager*
In the context of (role):	Providing top-level guidance on how data is structured
Motivation:	
At company X, data rules are divided between top-level (also: reference) ontologies and domain-level ontologies. The top-level ontologies provide general building blocks to structure data and layout rules to which all or most instances must comply.	

Table 8.
ROMEO: Example Question for SHACLEval.

Questions for SH_Top1: Usage Level of top-level ontologies.	
SH_Top1_Q1	Are the top-level ontologies adequately used to constrain the data?
Discussion:	
Top-level ontologies are meant to be used organization-wide. If they are applied only to a small subset of data, that is a potential problem, as most data does not adhere to self-set structures, which can hinder reusability.	

Table 9.
ROMEO: Example Metrics for SHACLEval

Measurements for SH_Reference1: Usage Level of top-level ontologies.	
SH_Top1_Q1_M1:	Metric: $ShapeEfficiency$
	Desirable *Value Range: > 150*
	Tendency: *Must increase*
SH_Top1_Q1_M2:	Metric: $Ns_{indirect}(cls)Ratio$
	Tendency: *Should Increase*

Discussion:
The $ShapeEfficiency$ indicates how many shapes are constrained by a set of shapes. As our goal is a high usage level of our top-level ontologies, the usage level of these shapes should increase monotonously, and every top-level ontology should be applied to at least 150 instances.
The $Ns_{indirect}(cls)Ratio$ tells us how much of the data is applied to classes by inheritance. An increase in this metric indicates that the hierarchical, thus, the distinction of domain- and top-level ontologies to order data is actively used.

4. The Bosch Perspective on SHACL and its Evaluation

The SHACLEval development was motivated primarily by the industrial need to understand how the developed SHACL validations evolve. Bosch increasingly builds KGs to interlink the internal data. With the increase in size and use cases, the demands to validate that the data is developing according to the given needs were also raised.

4.1. The Motivation for Bosch to Use SHACL in Practice

Whereas OWL used to be the language of choice for expressing axioms and constraints, this has fundamentally changed with the occurrence of SHACL over the past years. At Bosch, we have experienced a shift from using OWL ontologies and their foundations in description logic towards ontologies more and more relying on SHACL instead. There are several reasons. First, OWL and its open world assumption (OWA) does not fit well with industrial use cases in general. The information in industrial KGs must be complete to ensure the proper functioning of the related use cases, applications, or products.

The OWA assumes an open world, where some facts are available and accessible, while others are stored at some other place(s) and might currently not be available. This worldview conveys that it is impossible to conclude something from the absence of a fact, such as the fact does not exist and is not valid. In general, the OWA leads to several complications when trying to validate KGs with OWL reasoning. Although OWL has means for defining minimum and maximum cardinalities of properties (owl:minCardinality, owl:maxCardinality), OWL reasoning is unsuitable for detecting cardinality violations.

First, violated minimum cardinality constraints do not lead to any error, as due to the OWA, the respective instance might have some value(s) for the respective property, but the facts are not available right now. Second, OWL behaves much differently for maximum cardinality constraints. An individual having two values for a property with a defined maximum cardinality of 1 does not cause an error either but instead results in the inference that the two individuals must be the same (owl:sameAs inference). A reasoner would report an error only by defining disjointness axioms between (all) classes of an ontology (owl:disjointWith), which causes a combinatorial complexity. Such an error reports a contradiction of the owl:sameAs inference with the disjointness of the two individuals' classes, but not the maximum cardinality constraint violation, which initially caused it. This behavior is unintuitive for users, who need to understand the root cause quickly to fix and resolve it.

Another vital reason at Bosch is the extensive requirements for ontology creation. Few semantic experts are available to perform such a vast task while ensuring high quality. With more than 500 ontologies developed, a SHACL-based framework like the one proposed in this paper is of core relevance to maintaining the quality of the ontologies. Furthermore, these ontologies are developed following three levels of ontology specifications. The three levels are Top, Domain, and Application ontologies. The aim here is to enable reusability across different divisions and foster standardization. To that end, it is also required to enable the automatic checking of the process of reusing ontologies properly.

SHACL is a better fit for the characteristics of industrial use cases that we see at Bosch. Its closed world assumption (CWA) allows for checking minimum and maximum cardinality constraints (sh:minCount, sh:maxCount), amongst many other things. Detected violations are well described by a SHACL engine, comprising an explanation and links to the particular ontology entities. This intuitive behavior helps users to understand and resolve the issues quickly.

In addition, SHACL constraints also play a crucial role in the creation of KGs at Bosch. Typically, the data that are utilized to create KGs come from different silos, like, relational databases, JSON, or CSV. The required transformation from these silos to the KGs by using the available ontologies typically suffers from inconsistencies due to the required transformations to create the KGs. For instance, the number of manufacturing lines may differ between the source data that was used to build the KG and the KG itself after the typical ETL process. To that end, SHACL constraints ensure that the data in the KG presented to final users remains aligned with the one available in the sources.

4.2. A Case for Analyzing SHACL in Practice.

We have started the application of the SHACLEval evaluation on actual use cases, e.g., the Line Information System (LIS) [28]. LIS is a KG-based solution that semantically harmonizes and integrates manufacturing data. LIS enables different use cases in the manufacturing context while resolving semantic conflicts from different data sources, e.g., Enterprise Resource Planning (ERP) systems, Manufacturing Execution Systems (MES), and Master Data Systems (MD). Due to the constant improvement of requirements, it is paramount to check that the ontologies that LIS utilizes and the KG generated are of adequate quality to be used in the real world. In this context, validating these artifacts with SHACL is core to the approach.

Another use case is the Home Comfort KG. The KG contains data from Bosch Home Comfort, in particular semantic models of residential heating systems and heat pumps, including their components, hardware, firmware, and more. As shown in

Table 10, the KG has a size of 511K triples, comprising 209 classes, 315 properties, 189 SHACL node shapes, 534 SHACL property shapes, and 39,569 instances stored in an Apache Jena Fuseki triple store. The maintenance of the data, i.e., instances, relationships, and literal values, was handled by the Knowledge Graph Explorer, which is a user interface for KGs developed by Bosch [29]. The KG Explorer allows users to conveniently view, browse, search, and edit data in a KG. For the Home Comfort KG, a GIT version history over the past two years exists at Bosch, which was collected by an automatic GIT versioning service. With a frequency of one hour, any new changes (if any) were automatically committed to the GIT repository. Overall, 10,809 changes were saved in the GIT repository, with over 5 million updated triples in total.

Table 10.
Characteristics of the Bosch Home Comfort knowledge graph

KG	Cls	OP	Ns	Ps	Ind	$Triples$
Home Comfort	209	315	189	534	39,569	511K

The connections between quality and development processes are possible by applying the SHACLEval measurement framework as part of the NEOntometrics application. SHACLEval was developed in close coordination with the named use cases and allows the detection of commonly occurring pitfalls and their improvements over time. Common pitfalls include:

- **Top-Level-Domain Reuse:** The example is similar to the challenge described in section 3.3: Top-level shapes are only helpful if they are regularly used in the underlying domains. The level of indirect applications to classes with the measure $Ns_{indirect}(Cls)Ratio$ indicates how much of a concept defined by the knowledge engineering department is being picked up in the domain.

- **Missing Non-Validation-Information:** While the shapes' primary function is the structural validation of incoming data, they still need to deliver human-centered information on their function and usage context. The change of values like $Ns_{wNonVal}ToNs$ allows us to understand whether the ontology is improving in this regard or not.
- **Cardinality Disbalances:** Cardinality restrictions for properties define the number of outbound edges and are at the forefront of shaping the graph. The $Ps(Op)ToOp$ ratio shows how many of the measures do have cardinality restrictions. A decrease indicates that more object properties are introduced that are not restricted by SHACL. An increase in the $Ps_{min}Ratio$ measure indicates that more property shapes are introduced that have minimum cardinality restrictions but not a defined maximum, which indicates potentially missing value ranges.

5. Exemplary Analysis with SHACLEval on DCAT-AP

In our research, we faced restrictions on using enterprise data due to its proprietary nature and confidentiality concerns. We decided to leverage an open-source and publicly available dataset. This approach ensures compliance with data privacy regulations and promotes transparency and reproducibility in our findings, enabling the broader research community to validate and build upon our work.

The Data Catalog Vocabulary (DCAT) is a W3C standard[3] that facilitates interoperability between web-based data catalogs. It builds on established vocabularies, like prov, foaf, or skos. The initiative has been picked up especially by states to facilitate interoperability of public data. In the EU, the member states use DCAT-AP (Application Profile)[4], a subset of DCAT with stricter requirements that aim to make public sector data more accessible and reusable.

The sample analysis builds on an analysis of the validation tool for the Norwegian adoption of the DCAT-AP data catalog. It uses a customized, commercial variant of the Neontometrics application[5]. For the analysis, the distributed files were merged into one KG. The data and the corresponding analysis are available online[6]. The built tool allows using an API based on a SHACL validation engine and corresponding shapes. The corresponding graphs and the code are open source. The authors of this paper are not affiliated with the initiative. In that sense, the given analyses are illustrative.

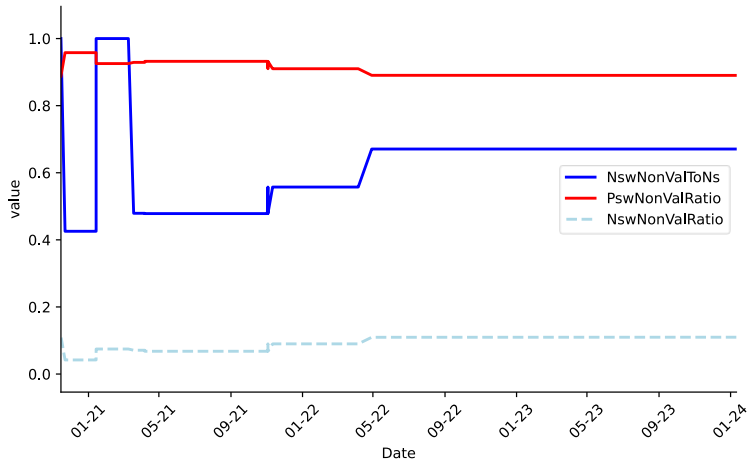

Figure 2. An Exemplary Analysis of the DCAT-AP SHACL shapes for the Norwegian validation engine.

[3] https://www.w3.org/TR/vocab-dcat-3/
[4] https://github.com/Informasjonsforvaltning/dcat-ap-no-validator-service
[5] http://neontometrics.com
[6] https://doi.org/10.5281/zenodo.16538766

Figure 2 shows the evolution of the measure $Ns_{wNonVal}ToNs$, $Ps_{wNonVal}Ratio$ an the inverse value $Ns_{wNonVal}Ratio$. It indicates that around half of the declared Ns do not have non-validational messages attached. There was a previous period where all Ns had human-centered information, but with an increase in NodeShapes, not all new shapes have non-validational information.

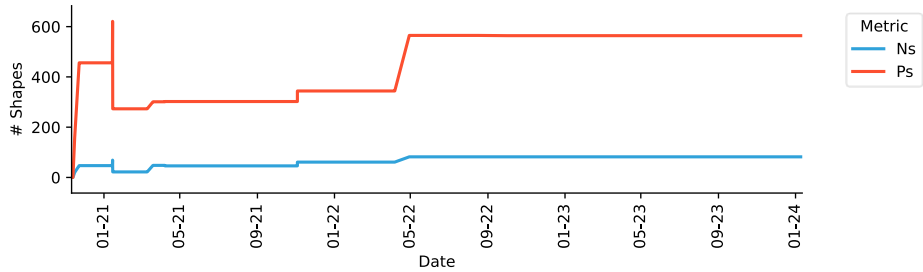

Figure 3. The number of explicitly defined Node- and Property shapes. Most of the Property Shapes are implicitly defined as part of a NodeShape.

Figure 3 further indicates a rise in property shapes. The combination of the first two diagrams tells a story of a growing graph where not all the newly created constraints have human-centered information attached.

Finally, Figure 4 evaluates the constraints on actual classes. The classes are primarily used in the given repository to build test cases. It reveals that (A) the number of class constraints by SHACL shapes increases over time. However, the number of restricted classes (mean: 26,24, cf. Figure 3) is relatively modest compared to the number of declared node shapes (mean 51.81, cf. Figure 2). Thus, it might indicate that there are gaps in the tests performed. (B) Further indicates that the validation rarely uses the RDFS entailment regime, as most constrained classes are targeted directly.

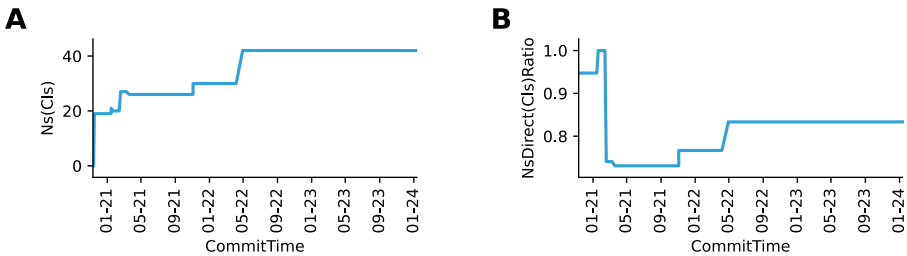

Figure 4. A: $Ns(Cls)$, the number of constrained classes, B: $Ns_{direct}(Cls)Ratio$, The ratio of directly constrained classes

6. Conclusion

SHACL quickly became an indispensable tool for validating graph data through structural constraints and is now often at the core of building practice-oriented knowledge graphs. Its focus on validation brings new challenges regarding quality management. In this work, we presented SHACLEval, a framework for evaluating SHACL-based ontologies and their usage on actual data or data structures.

SHACLEval addresses the validation specifics of SHACL. It proposes measures that allow knowledge engineers to quickly grasp the inner structure of the validation graph and its impact on the rest of the ontology and the data. An evolutionary analysis using SHACLEval can identify a potential drift of the data strategy and the actual knowledge graph developments.

The quality of SHACL shapes depends on the individual use cases and the goals of an artifact. A top-level shape has a different structure than shapes made for the domain. To establish a quality measurement with metrics, one must first identify the KPIs that measure the aspects necessary for the desirable attributes of a given graph. In that sense, the proposed measures are potentially reusable metrics, but the list is non-exhaustive. For example, the measurements in Table 1 can be reused and

combined, creating the measurements that best capture quality for the use case. To make the most value out of the metrics, more research is needed towards reasonable combinations of metrics and the informational value of their measurements when being combined. This could avoid the necessity to identify KPIs and related measures individually per use case, but to have a framework of predefined, well-understood sets of metrics and their specific informational value defined, from which users can choose from to easily get started.

We believe there is a solid need to measure graph constraints. The use cases of Bosch indicate that, on the one hand, SHACL is already being used extensively and increasingly replaces OWL. On the other hand, the rise in size, usage scenarios, and complexity emphasizes the necessity of understanding how the graph is evolving and whether self-set modeling goals are met.

Objective measures, like the ones proposed in the SHACLEval framework, allow graph structure breakdown into a simple number. This number can strengthen the quality by providing a bird's-eye perspective on the developments, showing potential improvements, and ensuring that the developments adhere to self-given standards.

In the future, we are planning to investigate existing methods on using LLMs for ontology engineering (e.g., [30,31] and evaluation (e.g., [32,33]), as a basis to develop LLM-enabled methodologies and tool support for SHACLEval framework adoption. Furthermore, we plan to scale the evaluation of SHACLEval through various research and industrial use cases.

Declaration on Generative AI

During the preparation of this work, the author(s) used Grammarly in order to: Grammar and spelling check, improve writing style. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] C. Gutierrez, J.F. Sequeda, Knowledge graphs, Commun. ACM 64 (2021) 96–104. https://doi.org/10.1145/3418294.

[2] S. Steyskal, K. Coyle, SHACL Use Cases and Requirements, (2017). https://www.w3.org/TR/2017/NOTE-shacl-ucr-20170720/ (accessed March 20, 2024).

[3] J. Raad, C. Cruz, A Survey on Ontology Evaluation Methods, in: A. Fred (Ed.), Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management: Lisbon, Portugal, November 12 - 14, 2015, SciTePress, Setúbal, 2015: pp. 179–186. https://doi.org/10.5220/0005591001790186.

[4] A.M. Porn, C.A.G. Huve, L.M. Peres, A.I. Direne, A systematic Literature Review of OWL Ontology Evaluation, in: P. Isaías, L. Rodrigues (Eds.), Proceedings of the 15th International Conference WWW/Internet 2016: Mannheim, Germany, October 28-30, 2016, IADIS Press, Lissabon?, 2016.

[5] R.S.I. Wilson, J.S. Goonetillake, W.A. Indika, A. Ginige, Analysis of Ontology Quality Dimensions, Criteria and Metrics, in: O. Gervasi, B. Murgante, S. Misra, C. Garau, I. Blečić, D. Taniar, B.O. Apduhan, A.M.A.C. Rocha, E. Tarantino, C.M. Torre (Eds.), Computational Science and Its Applications – ICCSA 2021, Springer International Publishing, Cham, 2021: pp. 320–337. https://doi.org/10.1007/978-3-030-86970-0_23.

[6] S. Tartir, I.B. Arpinar, Ontology Evaluation and Ranking using OntoQA, in: International Conference on Semantic Computing, 2007: ICSC 2007; 17 - 19 Sept. 2007, Irvine, California; Proceedings; [Held in Conjunction with] the First International Workshop on Semantic Computing and Multimedia Systems (IEEE-SCMS 2007), IEEE Computer Society, Los Alamitos, Calif., 2007: pp. 185–192. https://doi.org/10.1109/ICSC.2007.19.

[7] S. Tartir, I.B. Arpinar, M. Moore, A.P. Sheth, B. Aleman-Meza, OntoQA: Metric-Based Ontology Quality Analysis, in: D. Caragea, V. Honavar, I. Muslea, R. Ramakrishnan (Eds.), IEEE Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources, 2005.

[8] A. Gangemi, C. Catena, M. Ciaramita, J. Lehmann, A theoretical framework for ontology evaluation and validation, in: P. Bouquet, G. Tummarello (Eds.), Semantic Web Applications and Perspectives, CEUR, 2005. http://ceur-ws.org/Vol-166/.
[9] A. Gangemi, C. Catenacci, M. Ciaramita, J. Lehmann, R. Gil, F. Bolici, Strignano Onofrio, Ontology evaluation and validation: An integrated formal model for the quality diagnostic task, Trentino, Italy, 2005.
[10] A. Duque-Ramos, J.T. Fernández-Breis, R. Stevens, N. Aussenac-Gilles, OQuaRE: A square-based approach for evaluating the quality of ontologies, Journal of Research and Practice in Information Technology 43 (2011) 159–176.
[11] A. Reiz, K. Sandkuhl, A Critical View on the OQuaRE Ontology Quality Framework, in: J. Filipe, M. Śmiałek, A. Brodsky, S. Hammoudi (Eds.), Enterprise Information Systems, Springer Nature Switzerland, Cham, 2023: pp. 273–291. https://doi.org/10.1007/978-3-031-39386-0_13.
[12] A. Fernández-Izquierdo, M. Poveda-Villalón, A. Gómez-Pérez, R. García-Castro, Towards metrics-driven ontology engineering, Knowl Inf Syst 63 (2021) 867–903. https://doi.org/10.1007/s10115-021-01545-9.
[13] H. Yao, A.M. Orme, L. Etzkorn, Cohesion Metrics for Ontology Design and Application, J. of Computer Science 1 (2005) 107–113. https://doi.org/10.3844/jcssp.2005.107.113.
[14] Y. Ma, B. Jin, Y. Feng, Semantic oriented ontology cohesion metrics for ontology-based systems, Journal of Systems and Software 83 (2010) 143–152. https://doi.org/10.1016/j.jss.2009.07.047.
[15] S. Oh, H.Y. Yeom, J. Ahn, Cohesion and coupling metrics for ontology modules, Inf Technol Manag 12 (2011) 81–96. https://doi.org/10.1007/s10799-011-0094-5.
[16] A. Zaveri, A. Rula, A. Maurino, R. Pietrobon, J. Lehmann, S. Auer, Quality assessment for Linked Data: A Survey, Semant Web 7 (2015) 63–93. https://doi.org/10.3233/SW-150175.
[17] A.C. Junior, J. Debattista, D. O'Sullivan, Assessing the Quality of R2RML Mappings, (n.d.).
[18] J. Debattista, S. Auer, C. Lange, Luzzu—A Methodology and Framework for Linked Data Quality Assessment, Journal of Data and Information Quality 8 (2016) 1–32. https://doi.org/10.1145/2992786.
[19] H. Knublauch, D. Kontokostas, Shapes Constraint Language (SHACL), (2007). https://www.w3.org/TR/shacl/.
[20] P. Pareti, G. Konstantinidis, A Review of SHACL: From Data Validation to Schema Reasoning for RDF Graphs, in: M. Šimkus, I. Varzinczak (Eds.), Reasoning Web. Declarative Artificial Intelligence, Springer International Publishing, Cham, 2022: pp. 115–144. https://doi.org/10.1007/978-3-030-95481-9_6.
[21] K. Rabbani, M. Lissandrini, K. Hose, SHACTOR: Improving the Quality of Large-Scale Knowledge Graphs with Validating Shapes, in: Companion of the 2023 International Conference on Management of Data, ACM, Seattle WA USA, 2023: pp. 151–154. https://doi.org/10.1145/3555041.3589723.
[22] S. Lieber, B.D. Meester, A. Dimou, R. Verborgh, Statistics about Data Shape Use in RDF Data, in: Proceedings of the ISWC 2020 Demos and Industry Tracks: From Novel Ideas to Industrial Practice Co-Located with 19th International Semantic Web Conference (ISWC 2020), Online, 2020. https://ceur-ws.org/Vol-2721/paper584.pdf.
[23] A. Cimmino, A. Fernández-Izquierdo, R. García-Castro, Astrea: Automatic Generation of SHACL Shapes from Ontologies, in: A. Harth, S. Kirrane, A.-C. Ngonga Ngomo, H. Paulheim, A. Rula, A.L. Gentile, P. Haase, M. Cochez (Eds.), The Semantic Web, Springer International Publishing, Cham, 2020: pp. 497–513. https://doi.org/10.1007/978-3-030-49461-2_29.
[24] B. Spahiu, A. Maurino, M. Palmonari, Towards Improving the Quality of Knowledge Graphs with Data-driven Ontology Patterns and SHACL, (n.d.).
[25] M. Rashid, M. Torchiano, G. Rizzo, N. Mihindukulasooriya, O. Corcho, A quality assessment approach for evolving knowledge bases, Semant Web 10 (2019) 349–383. https://doi.org/10.3233/SW-180324.
[26] J. Yu, J.A. Thom, A. Tam, Requirements-oriented methodology for evaluating ontologies, Information Systems 34 (2009) 766–791. https://doi.org/10.1016/j.is.2009.04.002.
[27] V. Basili, G. Caldiera, H.D. Rombach, The Goal Question Metric Approach, (n.d.).
[28] I. Grangel-González, M. Rickart, O. Rudolph, F. Shah, LIS: A Knowledge Graph-Based Line Information System, in: C. Pesquita, E. Jimenez-Ruiz, J. McCusker, D. Faria, M. Dragoni, A. Dimou, R. Troncy, S. Hertling (Eds.), The Semantic Web, Springer Nature Switzerland, Cham, 2023: pp. 591–608. https://doi.org/10.1007/978-3-031-33455-9_35.

[29] H. Dibowski, Enhancing the viewing, browsing and searching of knowledge graphs with virtual properties, IJWIS (2024). https://doi.org/10.1108/IJWIS-02-2023-0027.
[30] H. Babaei Giglou, J. D'Souza, S. Auer, LLMs4OL: Large Language Models for Ontology Learning, in: T.R. Payne, V. Presutti, G. Qi, M. Poveda-Villalón, G. Stoilos, L. Hollink, Z. Kaoudi, G. Cheng, J. Li (Eds.), The Semantic Web – ISWC 2023, Springer Nature Switzerland, Cham, 2023: pp. 408–427. https://doi.org/10.1007/978-3-031-47240-4_22.
[31] B. Zhang, V.A. Carriero, K. Schreiberhuber, S. Tsaneva, L.S. González, J. Kim, J. De Berardinis, OntoChat: A Framework for Conversational Ontology Engineering Using Language Models, in: A. Meroño Peñuela, O. Corcho, P. Groth, E. Simperl, V. Tamma, A.G. Nuzzolese, M. Poveda-Villalón, M. Sabou, V. Presutti, I. Celino, A. Revenko, J. Raad, B. Sartini, P. Lisena (Eds.), The Semantic Web: ESWC 2024 Satellite Events, Springer Nature Switzerland, Cham, 2025: pp. 102–121. https://doi.org/10.1007/978-3-031-78952-6_10.
[32] N. Tufek, A.S. Thuluva, V.P. Just, F.J. Ekaputra, T. Bandyopadhyay, M. Sabou, A. Hanbury, Validating Semantic Artifacts with Large Language Models, in: A. Meroño Peñuela, O. Corcho, P. Groth, E. Simperl, V. Tamma, A.G. Nuzzolese, M. Poveda-Villalón, M. Sabou, V. Presutti, I. Celino, A. Revenko, J. Raad, B. Sartini, P. Lisena (Eds.), The Semantic Web: ESWC 2024 Satellite Events, Springer Nature Switzerland, Cham, 2025: pp. 92–101. https://doi.org/10.1007/978-3-031-78952-6_9.
[33] S. Tsaneva, S. Vasic, M. Sabou, LLM-driven Ontology Evaluation: Verifying Ontology Restrictions with ChatGPT, (n.d.).