

Towards SHACL-based Knowledge Graph Transformation of Visual Domain Knowledge

Stefan Bischof¹, Erwin Filtz¹, Josiane Xavier Parreira¹, Simon Steyskal¹, Michael Baumgart², David Gruber², Maximilian Liebetreu², Florian Rötzer² and Stephan Strommer²

¹Siemens AG Österreich, Vienna, Austria

²AIT Austrian Institute of Technology GmbH, Center for Vision, Automation & Control, Vienna, Austria

Abstract

Effective knowledge representation plays a pivotal role in harnessing the full potential of domain-specific information. Through tools like Infinity Maps, domain knowledge can be easily captured in a visual manner. However, translating these visually intuitive representations to formal, machine-processable formats often necessitates expert knowledge, thereby creating a significant barrier between domain experts and knowledge engineers. While domain experts possess deep understanding of their respective domains, they often lack the formalisation skills required to transform this knowledge into machine-readable formats. Conversely, knowledge engineers can design and implement sophisticated knowledge graphs, but may not have access to the domain-specific expertise necessary for effective knowledge representation. To address this challenge, we propose a novel approach that leverages SHACL (Shape Constraint Language) rules to transform visual domain knowledge expressed as Infinity Maps into knowledge graphs. Our method enables domain experts to define their knowledge structures using familiar Infinity Map representations, which are then transformed into standardised knowledge graphs compliant with the SHACL standard.

Keywords

Knowledge Graphs, Infinity Maps, RDF, SHACL, Semantic Web

1. Introduction and Motivation

The setup and optimisation of industrial production processes, such as, for example, high-pressure die casting, heavily rely on the expert knowledge and experience of a few individuals within a company. Consequently, domain knowledge often remains personal property rather than a shared company asset, creating a dependency on specific personnel. Despite companies' quality management standards, this valuable knowledge is frequently undocumented and undigitised, making it inaccessible to the broader workforce. This lack of effective knowledge management and transfer hinders resource-efficient, green production of advanced products,

SEMANTICS 2024

✉ bischof.stefan@siemens.com (S. Bischof); erwin.filtz@siemens.com (E. Filtz); josiane.parreira@siemens.com (J. X. Parreira); simon.steyskal@siemens.com (S. Steyskal); michael.baumgart@ait.ac.at (M. Baumgart); david.gruber@ait.ac.at (D. Gruber); maximilian.liebetreu@ait.ac.at (M. Liebetreu); florian.roetzer@ait.ac.at (F. Rötzer); stephan.strommer@ait.ac.at (S. Strommer)

ORCID 0000-0001-9521-8907 (S. Bischof); 0000-0003-3445-0504 (E. Filtz); 0000-0002-3050-159X (J. X. Parreira); 0000-0002-5183-2486 (S. Steyskal); 0009-0009-6776-4404 (M. Baumgart); 0000-0002-7544-5632 (D. Gruber); 0000-0001-8374-8476 (M. Liebetreu); 0000-0003-2661-5575 (F. Rötzer); 0009-0009-9349-9683 (S. Strommer)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

such as complex die-casting parts, especially in times of skilled labour shortages. It has therefore become a pressing issue for industries to find a low-threshold, minimal-effort way for experts to digitise and share their knowledge, making it machine-readable, accessible and usable also for non-domain experts. A toolchain that achieves this goal should include a highly accessible tool for experts to document their domain knowledge. Examples of such tools are Conceptboard ¹, Microsoft Whiteboard ², and Infinity Maps ³.

In the present paper, we will use Infinity Maps due to their rich JSON exporting capabilities. Additionally, a technology and/or database that is able to store knowledge in a structured form is of utter importance. In our case, the output of the toolchain is a knowledge graph (KG) in RDF format. The envisaged toolchain is outlined in Fig. 1.

Contribution. We design a transformation pipeline from Infinity Maps to a structured RDF graph, based on the SHACL standard⁴. Features include: (i) The authors of an Infinity Map are free to dump a mixture of structured and unstructured knowledge and data into each Map. (ii) Authors do not require any technical understanding of the KG that will be produced. (iii) By employing SHACL, authors of an Infinity Map receive automated assistance in structuring the domain knowledge for computational processing. (iv) Using SHACL, the generated KG is guaranteed to conform to the requirements of any ontologies applied to the graph.

2. From Domain Knowledge to Knowledge Graph

Step 1: Visual modelling by schema. The first step in our pipeline is to organise and gain an overview of all available expert and domain knowledge, which comes in various formats like PNG, PDF, CSV files, emails, and interviews. Infinity Maps excels in visualising and organising this unstructured information. It allows the creation and connection of cards into tree-like hierarchies, a feature we use extensively to efficiently manage and structure our knowledge base. Once an overview on all available knowledge sources has been attained, Infinity Maps can also be used to combine and organise what information was learned from these sources. To this end, we designed a loose schema based on the gathered information and visually modelled this domain knowledge accordingly.

Remark. *The availability of visual tools for structured KG creation is limited. This is unsurprising: KGs are a comparatively novel concept, and while they have had a lot of success in recent years, most success stories originate from fields and applications that create such KGs automatically from other forms of structured databases, with the main challenge being leveraging the information contained within (triple prediction for recommender systems, etc.) [1, 2, 3]. However, in industrial production domains, the degree of digitisation is often surprisingly low, with knowledge being available in hand-written form, separate documents, literature, and the minds of experts and operators drawing from said literature and their own experience. Structuring this knowledge in any form, but particularly in an explainable way and one that can be easily and efficiently queried, is a vital step towards leveraging all available knowledge to optimise processes and products.*

Step 2: Visual modelling by ontology. In the later stages of visual modelling, we developed an ontology for the envisaged KG, thereby clarifying the modelling guidelines. We utilised tags and tree-like hierarchies within Infinity Maps to denote relationships between entities and label

¹Conceptboard: <https://conceptboard.com/>

²Microsoft Whiteboard: <https://www.microsoft.com/en-us/microsoft-365/microsoft-whiteboard>

³Infinity Maps: <https://infinitymaps.io/>

⁴SHACL: <https://www.w3.org/TR/shacl>

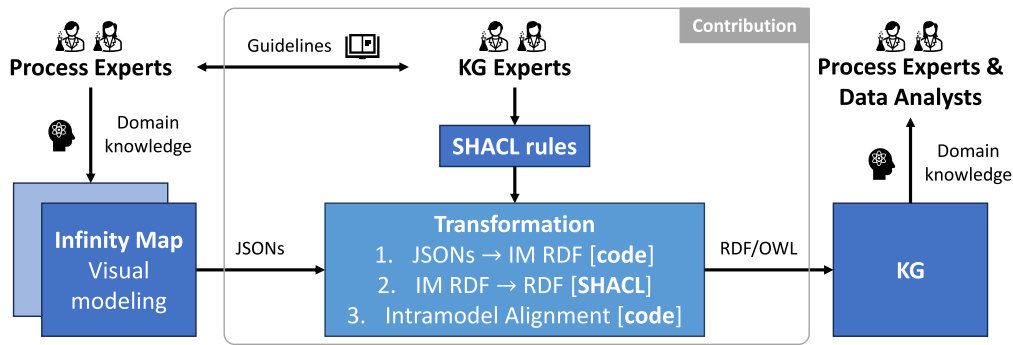


Figure 1: Transformation of domain knowledge to knowledge graph using visual modelling with Infinity Maps and SHACL.

cards accordingly. These tags and hierarchical logic subsequently guides the transformation process from Infinity Maps to the KG.

A significant drawback of Infinity Maps is the absence of dynamic links between cards. Although each card has a unique URL and can be referenced via hyperlinks, these links are static text and can easily break during the Map authoring process. To maintain simplicity for human readability, we opted to cross-reference tagged cards by ensuring that each combination of card label and tag remains unique throughout the entire Infinity Maps project.

Due to this limitation in dynamic linking, we anticipate the appearance of duplicate entities and errors in triplets when converting from Infinity Maps to a KG. This necessitates additional constraints and rules for a successful transformation.

Step 3: Transformation by constraints. Infinity Maps allows exporting each Map to JSON format. We transform these JSON structures to a KG using Python code and shapes implemented in SHACL. This procedure is described in much more detail in Section 3.

Closing the loop. From the KG, new domain knowledge can be gained by experts. Any new knowledge can be added to the Infinity Maps and the KG itself, thereby closing the loop. An overview of the entire pipeline is given in Fig. 1.

3. SHACL-based Transformation Framework

The Shapes Constraint Language (SHACL), is a W3C recommendation designed for validating RDF graphs against a set of SHACL shapes, i.e. the constraints the to-be-validated RDF graph has to adhere to. Such constraints can include (but are not limited to), e.g., checking existence of particular properties, data types, value ranges, and relationships between nodes⁵. Using SHACL, one can ensure that the data conforms to the expected structure and semantics, enabling reliable data integration and interoperability.

Additionally, we utilise SHACL Rules⁶ to facilitate the transformation of raw data (i.e., the Infinity Maps JSON exports) into a structured, and semantically enriched representation that aligns with predefined ontologies and the domain understanding as provided by the domain experts.

⁵SHACL Core Components: <https://www.w3.org/TR/shacl/#core-components>

⁶SHACL Rules were introduced as part of the SHACL Advanced Features Note: <https://www.w3.org/TR/shacl-af>.

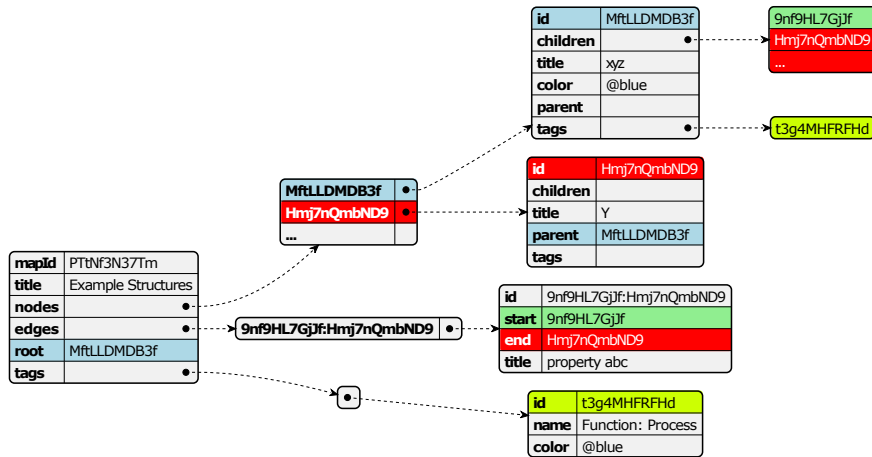


Figure 2: Excerpt of an Infinity Maps JSON export.

3.1 Handling Infinity Maps Data As shown in Fig. 2, an Infinity Maps JSON export follows a very basic structure. At its core, each Infinity Map is represented as a JSON object with three main elements: *nodes*, *edges*, and *tags*. Where *nodes* represent all nodes in the Map, *edges* all edges between nodes, and *tags* are all tags used in the Map. As depicted in Listing 1, each node has a unique identifier, a title, and optionally a reference to its parent node, and a list of references to any of its child nodes. Each edge has a source and target node, and a name. Each tag has a unique identifier and a title⁷.

3.2 Enrichment with Domain-Specific SHACL Rules

Based on modeling guidelines specified by the process/domain experts, we define SHACL rules that capture the unique semantics and requirements of the domain. For example, one of the guidelines states that relations are represented by chaining at least two parent-child relationships between a starting entity and one or more target entities, where a relation contains exactly one node in its path that defines the type of the relation. The SHACL rule in Listing 2 captures this by searching for paths starting from a set of focus nodes of type `dg:Quantity`, traversing through one or more `im:child` relationships to intermediate nodes `?p`, and finally reaching target entities `?mid`. Using the `VALUES` clause, we define the properties to be used based on the labels of the intermediate nodes. For example, for the example triples in Listing 1, the following triple would

```

ex:FH9d47H93gf a dg:KPI, im:Node ;
rdfs:label "Schließkraft Err" ;
im:child ex:LnrDTfHTHqF ;
im:id "FH9d47H93gf" ;
im:parent ex:99dqJd2rjLb ;
im:tag ex:Jj3qLGjGhGp ;
im:title "Schließkraft Fehler" .

ex:LnrDTfHTHqF a im:Node ;
rdfs:label "Abhängigkeiten" ;
im:child ex:P2jBpjh8RjM, ex:Qbd7rnb ;
im:id "LnrDTfHTHqF" ;
im:parent ex:FH9d47H93gf ;
im:title "Abhängigkeiten" .

ex:P2jBpjh8RjM a dg:Quantity, im:Node ;
rdfs:label "Schließkraft Err (Gießen)" ;
im:id "P2jBpjh8RjM" ;
im:parent ex:LnrDTfHTHqF ;
im:tag ex:BLJD8RqhTrd ;
im:title "Schließkraft Err (Gießen)" .

```

Listing 1: KPI that has a Quantity as dependency.

⁷Due to space limitations, we have not included sample triples for tags or edges.

be generated: `ex:FH9d47H93gf dg:has_dependency ex:P2jBpjh8RjM` .

```
enr:QuantityRule a sh:SPARQLRule ;
sh:construct """
  CONSTRUCT {
    $this ?rel ?mid .
  } WHERE {
    $this im:child ?typ .
    ?typ rdfs:label ?l ; im:child* ?p .
    ?p im:child ?mid .
    ?mid im:tag ?tag ; a ?target .
    VALUES (?l ?rel ?target) {
      ( "Abhängigkeiten" dg:has_dependency dg:Quantity)
      ( "Messung" dg:has_measurement dg:Signal)
    }
  }
  """ ;
sh:condition [ # evaluate rule only if focus node is a dg:Quantity
sh:property [
  sh:path rdf:type ;
  sh:hasValue dg:Quantity ;
] ;
] ; sh:prefixes <http://siemens.com/dgassist/enrichment> .
```

Listing 2: SHACL Rule for creating relations based on guidelines provided by domain experts.

Conclusion and Future Work In this paper, we presented a novel approach that enables domain experts to model their knowledge using an easy and intuitive visual representation, which is then exported as JSON, and afterwards transformed into a semantically enriched KG representation using SHACL. Future work will focus on integration of additional SHACL rules as well as evaluation of the transformation process on different real-world use cases.

Acknowledgments

This work was conducted within the Austrian research project DG Assist (FFG project number: FO999899053). This project is funded by the Federal Ministry for Climate Protection, Environment, Energy, Mobility, Innovation and Technology, BMK, and is carried out as part of the Production of the Future programme.

References

- [1] A. Hogan, et al., Knowledge graphs, *ACM Comput. Surv.* 54 (2021). URL: <https://doi.org/10.1145/3447772>. doi:10.1145/3447772.
- [2] J. Liu, L. Duan, A survey on knowledge graph-based recommender systems, in: *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 2021, pp. 2450–2453. doi:10.1109/IAEAC50856.2021.9390863.
- [3] N. Noy, Y. Gao, A. Jain, A. Narayanan, A. Patterson, J. Taylor, Industry-scale knowledge graphs: lessons and challenges, *Commun. ACM* 62 (2019) 36–43. URL: <https://doi.org/10.1145/3331166>. doi:10.1145/3331166.