SHACL Shape Extraction for Evolving Knowledge Graphs using QSE

Diploma thesis - Proposal

Eva Pürmayr, 11807199

March 2024

Supervision: Univ.Prof. Dr.-Ing. Dipl.-Inf. Katja Hose

1 Problem Statement

Storing complex and interconnected data is quite a challenge for data engineers. Knowledge graphs offer a versatile solution and have a broad range of applications, both in industry and in academia. Data quality is vital and is the basis for good results later on. As a validation language, SHACL can be used for RDF [1]. Previously, an approach called QSE (quality shape extraction), which automatically extracts SHACL shapes from large datasets, was released [2]. The user can define two parameters, which omit less useful shapes: support and confidence. Based on these generated node and property shapes, existing data can be validated. QSE is available as command-line tool, therefore Shactor was developed, a web-based tool that visualizes the extraction process and provides useful statistics [3].
Since knowledge graphs are not static, there may exist different versions of a graph, maybe with minimal changes. QSE and Shactor are specialized on shapes extraction but do not compare them between different versions of a graph. This work aims at shapes extraction and comparison, specifically with evolving knowledge graphs. As there is currently no tool available for this use case, the characteristics of usability and speed remain unmeasurable. Using manual methods is the only option, but this is tedious and lacks practicality. After this thesis is finished, users should be able to conveniently and efficiently compare SHACL shapes across various versions of a knowledge graph.

2 Goals and Expected Outcome

The main goal of this thesis is to develop a usable, web-based tool that allows shapes extraction and comparison for different versions of a graph. Some visual features will be based on Shactor. However, users can create multiple shape extractions with different parameters and compare them with previously generated shapes. The web-app shows which shapes stayed the same, which were added and which were deleted. Also, there will be information provided why shapes were included or removed.
The second part of the thesis deals with adaptions of the QSE algorithm. Using the terms V1 and V2 for two versions of a graph and the corresponding SHACL shapes generated by QSE (S1 and S2), an approach is to use the changeset between V1 and V2 during shape extraction so that S2 can be generated without using V2.
Another improvement in this direction is to create S2 and simultaneously

create the changeset between V1 and V2.
Furthermore, an algorithm similar to RQ1.2, which graphically explains why shapes from S1 have been added or deleted in S2, will be developed. This first approach uses support and confidence information provided by QSE in S1 and S2. However, there is also the option to create partial SPARQL queries based on S1 and execute them on V2 to reevaluate support and confidence. This approach can also determine which shapes from S1 will remain in S2. Advantages offered by this method are that QSE does not need to be applied to V2 and that it is sufficient for V2 to be available as a SPARQL endpoint. The web application will also provide graphical interfaces to use these methods. The most important characteristic of these algorithms is the execution time compared to the baseline, which is defined in the evaluation section.
For testing, different datasets will be used, where graph versions have different degrees of change. Datasets for this use case could be the ones used in BEAR [4].
The success criteria for this project involve the creation of a website enabling users to easily compare SHACL shapes and correctly developing the three algorithms mentioned.

3 Research Questions

Given two versions of a graph (V1 and V2) and the corresponding SHACL shapes generated by QSE (S1 and S2):

RQ1.1: What is an appropriate way to compare S1 and S2 in a user-friendly way?

RQ1.2: What is an appropriate way to explain why certain shapes from S1 have been added, removed or changed by using information from the QSE algorithm?

RQ2.1: Given the changeset between V1 and V2, what is an appropriate way to adapt the QSE algorithm so that it calculates S2 by using V1 and the changeset?

RQ2.2 What is an appropriate way to adapt the QSE algorithm so that it calculates S2 and simultaneously creates the changeset between V1 and V2?

RQ2.3: What is an appropriate way to explain why certain shapes of S1 have been added or removed in V2 by using partial SPAQRL queries?

Appropriate for RQ1.1 and RQ1.2 means user-friendly, useful and correct. It can be measured during the evaluation of semi-structured interviews with experts. Appropriate for RQ2.1, RQ2.2 and RQ2.3 means correct and faster in comparison to the method described in the evaluation section.

4 Research Methods

Design Science Research is selected as a scientific method [5]. In the relevance cycle as well as in the rigor cycle a **systematic literature review** will be used so that business needs, can be derived from the literature and knowledge from the rigor cycle can influence the thesis [6]. Ultimately, this process contributes to the knowledge base and aligns with evolving business needs by the rigor and relevance cycle's conclusion [7].
However, given the time-consuming nature of a SLR, only a partial review will be done at the beginning of the thesis. Only one online library (Google Scholar) will be used. The review protocol will contain the following items: objectives (e.g. to establish a broad understanding of the current knowledge base in the area of evolving knowledge graphs), search keywords (e.g. "evolving knowledge graphs"), study selection criteria (e.g. published after 2000), study exclusion criteria and procedure.
During the design cycle, **RQ1.1 and RQ1.2** will be answered using the **prototyping** method within the **construct phase** [8]. Initial stages will involve low-fidelity prototypes to determine the most suitable design that satisfies the end user's requirements. Iteratively, the minimum viable product will be developed, incorporating feedback from the evaluation phase.
Contrastingly, **RQ2.1, RQ2.2 and RQ 2.3** will use **algorithmic design** in the construct phase of the design cycle [9]. The approach entails maintaining algorithm simplicity, reusability, and utilizing libraries. Iterative development through small experiments will refine the algorithms. As realistic input, there will be graph snapshots from DBpedia and other data sources. During development, synthesized data and smaller versions of these graphs will be used.

5 Evaluation

For the evaluation part of the design cycle for **RQ1.1 and RQ1.2**, **semi-structured expert interviews** with students, who have already participated in Introduction To Semantic Systems will be conducted [10]. The impor-

tant part here is to measure the usability of the web-app in comparison to finding differences across the shapes without the tool. The interviews will contain a demonstration of the tool's functionality. Open-ended questions will be used to get comprehensive insights about usability. Since this method is time-consuming, 3-5 experts will be interviewed. In the planning phase, the interview guide will be created with prioritized questions, keeping the length of the interview as short as possible. The format of the interviews, online or offline, will be based on the preferences of the experts. It is planned to record the interviews.
The analysis of interview content will employ the inductive approach of **qualitative content analysis** [11]. Depending on the content of the interviews, categories will be created. Some of the feedback can then be implemented in the web-app.
The state-of-the-art solution would be to download the extracted shapes from Shactor and compare them with text-comparison tools.
For the evaluation phase in the design cycle for **RQ2.1, RQ2.2 and RQ 2.3**, **technical experiments** will be conducted, aligning with common practices in DSR projects for instantiations [12][13]. The experiments will be run on a virtual machine, since the data size of the test data will be huge. After it has been ensured, that the algorithm works correctly, speed will become the measurable metric.
The baseline here is the time it takes to generate S1 by using V1, plus the generation time of S2 by using V2. Additionally, a script will be written that compares S1 and S2 by their shape names. The execution time of this script will be added to the baseline.

6 State of the Art

To briefly explain the main reference of this project, QSE involves two iterations through all entities. During the initial phase (entity extraction), all instances based on their type declarations are counted. Subsequently, in the second run (entity constraints extraction), the algorithm gathers metadata for property shapes. Following this, support and confidence metrics are computed and finally, in the last step, the algorithm generates SHACL shapes.
The current solution for users to find similarities and differences between shapes would be to use Shactor for both versions, download the shapes and compare them manually via a text-comparison tool.
Regarding evolving knowledge graphs and defining versions of a graph in

general, there has already been a lot of research done [14] [15] [16] [17] [18]. GraphDB already offers a feature for data versioning and history [19].
However, in the context of this research project, adopting this form of versioning is currently not possible because not all graphs use the same form of versioning. Standardization across all graphs would be essential; otherwise, the application would only be applicable to specific graphs utilizing a particular versioning technique.
Another important topic, regardless of evolving knowledge graphs, is how data quality can be ensured in knowledge graphs. There has been some general research in this area [20] [21] [22] [23]. One way to enhance data quality, as it is used in QSE, is to automatically extract SHACL shapes from knowledge graphs. Examples are SHACLGEN or ShapeDesigner, which follow a similar approach as QSE-exact but lack the ability to process large knowledge graphs [2] [24] [25]. However, there exist even more algorithms, that automatically extract a schema from data, but have several other shortcomings [26] [27] [28] [29] [30] [31] [32].

7 Relevance to the Curriculum

This subject builds upon the Introduction to Semantic Systems course, which is part of the Information Systems Engineering Core curriculum [33]. A foundational understanding of semantic systems, specifically in RDF graphs and knowledge graphs, is required. It is also interesting to understand the application of knowledge graphs and the significance of their quality. The course Knowledge Graphs provides insights into these aspects [34]. In the context of RDF, familiarity with SPARQL and SHACL is important. Given that QSE is written in Java, proficiency in Spring and Web Development becomes essential. Expertise in Spring is covered in Distributed Systems Technologies [35]. Moreover, basic comprehension of usability and software project design (which is taught in Advanced Software Engineering) is valuable [36]. Addressing RQ2.1, RQ2.2 and RQ2.3, knowledge of algorithmic design is essential.

References

[1] *Shapes Constraint Language (SHACL)*, en, Jul. 2017. [Online]. Available: `https://www.w3.org/TR/shacl/` (visited on 11/28/2023).

[2] K. Rabbani, M. Lissandrini, and K. Hose, "Extraction of Validating Shapes from Very Large Knowledge Graphs," en, *Proceedings of the VLDB Endowment*, vol. 16, no. 5, pp. 1023–1032, Jan. 2023, ISSN: 2150-8097. DOI: 10.14778/3579075.3579078. [Online]. Available: https://dl.acm.org/doi/10.14778/3579075.3579078 (visited on 11/13/2023).

[3] ——, "SHACTOR: Improving the Quality of Large-Scale Knowledge Graphs with Validating Shapes," in *Companion of the 2023 International Conference on Management of Data*, ser. SIGMOD '23, event-place: Seattle, WA, USA, New York, NY, USA: Association for Computing Machinery, 2023, pp. 151–154, ISBN: 978-1-4503-9507-6. DOI: 10.1145/3555041.3589723. [Online]. Available: https://doi.org/10.1145/3555041.3589723.

[4] *BEAR — BEnchmark of RDF ARchives*. [Online]. Available: https://aic.ai.wu.ac.at/qadlod/bear.html (visited on 11/14/2023).

[5] A. Hevner, A. R, S. March, *et al.*, "Design Science in Information Systems Research," *Management Information Systems Quarterly*, vol. 28, pp. 75–, Mar. 2004.

[6] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering," vol. 2, Jan. 2007.

[7] A. Hevner, "A Three Cycle View of Design Science Research," *Scandinavian Journal of Information Systems*, vol. 19, Jan. 2007.

[8] M. Jesús, "Scientific Prototyping: A Novel Approach to Conduct Research and Engineer Products," Nov. 2018, pp. 32–35. DOI: 10.1109/ICSESS.2018.8663862.

[9] P. Sanders, "Algorithm Engineering – An Attempt at a Definition," in *Efficient Algorithms: Essays Dedicated to Kurt Mehlhorn on the Occasion of His 60th Birthday*, S. Albers, H. Alt, and S. Näher, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 321–340, ISBN: 978-3-642-03456-5. DOI: 10.1007/978-3-642-03456-5_22. [Online]. Available: https://doi.org/10.1007/978-3-642-03456-5_22.

[10] W. Adams, "Conducting Semi-Structured Interviews," in Aug. 2015, ISBN: 978-1-118-89360-9. DOI: 10.1002/9781119171386.ch19.

[11] P. Mayring, "Qualitative Content Analysis," *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research [On-line Journal], http://qualitative-research.net/fqs/fqs-e/2-00inhalt-e.htm*, vol. 1, Jun. 2000.

[12] J. Antony, "4 - A Systematic Methodology for Design of Experiments," in *Design of Experiments for Engineers and Scientists (Second Edition)*, J. Antony, Ed., Second Edition, Oxford: Elsevier, 2014, pp. 33–50, ISBN: 978-0-08-099417-8. DOI: https://doi.org/10.1016/B978-0-08-099417-8.00004-3. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780080994178000043.

[13] B. Durakovic, "Design of experiments application, concepts, examples: State of the art," *Periodicals of Engineering and Natural Sciences*, vol. 5, pp. 421–439, Dec. 2017. DOI: 10.21533/pen.v5i3.145.

[14] M. Frommhold, R. N. Piris, and N. Arndt, "Towards Versioning of Arbitrary RDF Data," en,

[15] A. Lohfink and D. McPhee, "Resource-Level Versioning in Administrative Geography RDF Data," en,

[16] M. Tasnim, D. Collarana, D. Graux, F. Orlandi, and M.-E. Vidal, "Summarizing Entity Temporal Evolution in Knowledge Graphs," en,

[17] R. Pernischova, D. Dell'Aglio, M. Horridge, M. Baumgartner, and A. Bernstein, "Toward Predicting Impact of Changes in Evolving Knowledge Graphs," Oct. 2019.

[18] M. U. Nuha, "Data Versioning for Graph Databases," en, 2019. [Online]. Available: https://repository.tudelft.nl/islandora/object/uuid%3A3cdbd161-3e6b-463a-957d-00ec0942917a (visited on 11/10/2023).

[19] *Data History and Versioning — GraphDB 10.4 documentation*. [Online]. Available: https://graphdb.ontotext.com/documentation/10.4/data-history-and-versioning.html?highlight=version (visited on 10/20/2023).

[20] *Tsaneva and Sabou - Hybrid Human-Machine Evaluation of Knowledge Graphs.pdf*. [Online]. Available: https://drive.google.com/file/u/0/d/1jzH1KhSn6UxxhayoZV5rCXQF1nsCI5a6/view?pli=1&usp=embed_facebook (visited on 11/10/2023).

[21] A. Schmickl, *5 steps to detect inconsistencies in evolving knowledge graphs*, en, Feb. 2021. [Online]. Available: https://alenaschmickl.medium.com/5-steps-to-find-inconsistencies-in-evolving-knowledge-graphs-6f3f88c0ab7b (visited on 11/10/2023).

[22] J. H. Brenas and A. Shaban-Nejad, "Proving the Correctness of Knowledge Graph Update: A Scenario From Surveillance of Adverse Childhood Experiences," *Frontiers in Big Data*, vol. 4, p. 660 101, May 2021, ISSN: 2624-909X. DOI: 10.3389/fdata.2021.660101. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8126660/ (visited on 11/10/2023).

[23] K. Kellou-Menouer, N. Kardoulakis, G. Troullinou, Z. Kedad, D. Plexousakis, and H. Kondylakis, "A survey on semantic schema discovery," *The VLDB Journal*, vol. 31, no. 4, pp. 675–710, Jul. 2022, ISSN: 0949-877X. DOI: 10.1007/s00778-021-00717-x. [Online]. Available: https://doi.org/10.1007/s00778-021-00717-x.

[24] I. Boneva, J. Dusart, D. Fernández Alvarez, and J. E. L. Gayo, *Shape Designer for ShEx and SHACL Constraints*, Published: ISWC 2019 - 18th International Semantic Web Conference, Oct. 2019. [Online]. Available: https://hal.science/hal-02268667 (visited on 11/29/2023).

[25] A. Keely, *Shaclgen: Shacl graph generator*. [Online]. Available: https://github.com/uwlib-cams/shaclgen (visited on 11/29/2023).

[26] N. Mihindukulasooriya, M. R. A. Rashid, G. Rizzo, R. García-Castro, O. Corcho, and M. Torchiano, "RDF shape induction using knowledge base profiling," en, in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, Pau France: ACM, Apr. 2018, pp. 1952–1959, ISBN: 978-1-4503-5191-1. DOI: 10.1145/3167132.3167341. [Online]. Available: https://dl.acm.org/doi/10.1145/3167132.3167341 (visited on 11/29/2023).

[27] D. Fernandez-Álvarez, J. E. Labra-Gayo, and D. Gayo-Avello, "Automatic extraction of shapes using sheXer," *Knowledge-Based Systems*, vol. 238, p. 107 975, 2022, ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2021.107975. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705121010972.

[28] B. Spahiu, A. Maurino, and M. Palmonari, "Towards Improving the Quality of Knowledge Graphs with Data-driven Ontology Patterns and SHACL," in *WOP@ISWC*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:52193280.

[29] *TopQuadrant — Enterprise Models for Data Governance*, en. [Online]. Available: https://www.topquadrant.com/ (visited on 11/29/2023).

[30] H. J. Pandit, D. O'Sullivan, and D. Lewis, "Using Ontology Design Patterns To Define SHACL Shapes," en,

[31] A. Cimmino, A. Fernández-Izquierdo, and R. García-Castro, "Astrea: Automatic Generation of SHACL Shapes from Ontologies," en, in *The Semantic Web*, A. Harth, S. Kirrane, A.-C. Ngonga Ngomo, *et al.*, Eds., vol. 12123, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2020, pp. 497–513, ISBN: 978-3-030-49460-5 978-3-030-49461-2. DOI: 10.1007/978-3-030-49461-2_29. [Online]. Available: http://link.springer.com/10.1007/978-3-030-49461-2_29 (visited on 11/29/2023).

[32] P. G. Omran, K. Taylor, S. R. Mendez, and A. Haller, "Towards SHACL Learning from Knowledge Graphs," en,

[33] *188.399 Introduction to Semantic Systems — TU Wien*. [Online]. Available: https://tiss.tuwien.ac.at/course/educationDetails.xhtml?dswid=9706&dsrid=233&semester=2023W&courseNr=188399 (visited on 11/28/2023).

[34] *192.116 Knowledge Graphs — TU Wien*. [Online]. Available: https://tiss.tuwien.ac.at/course/courseDetails.xhtml?dswid=9706&dsrid=155&courseNr=192116&semester=2023S (visited on 11/28/2023).

[35] *184.260 Distributed Systems Technologies — TU Wien*. [Online]. Available: https://tiss.tuwien.ac.at/course/courseDetails.xhtml?dswid=5007&dsrid=808&courseNr=184260&semester=2023S (visited on 11/28/2023).

[36] *180.456 Advanced Software Engineering — TU Wien*. [Online]. Available: https://tiss.tuwien.ac.at/course/courseDetails.xhtml?dswid=9706&dsrid=456&courseNr=180456&semester=2022W (visited on 11/28/2023).