rapid-triples: Adaptive Forms for Semi-automatic Knowledge Collection in RDF

Mario Scrocca*, Alessio Carenini, Valentina Anita Carriero and Irene Celino

Cefriel – Politecnico di Milano, Milan, Italy

Abstract

To reduce inaccuracies or a lack of context, AI applications may heavily benefit from structured knowledge modelled relying on reference ontologies. We present `rapid-triples`, a customizable and dynamic interface that facilitates human-in-the-loop collection of structured knowledge in RDF format. The system allows domain experts to manually input knowledge or semi-automatically enhance an extraction by an automated system from existing data sources. By utilizing a common schema mapped to a target ontology, *rapid-triples* ensures that the knowledge collected is semantically interoperable and machine-readable. This tool supports various use cases, including expert-guided knowledge creation, validating and refining outputs from automated extractors, and generating high-quality training data for AI systems. Finally, we discuss the adoption of `rapid-triples` to support a use case in the industrial domain through the collection and exploitation of procedures as knowledge graphs.

Keywords

knowledge collection, knowledge extraction, knowledge graph construction, form interface, human-in-the-loop

1. Introduction

The adoption of Large Language Models (LLMs) is reshaping the landscape of AI applications across different domains, yet their black-box nature often limits their ability to access and utilize factual knowledge, highlighting the need for a complementary integration with structured knowledge systems such as Knowledge Graphs (KGs) [1]. Capturing and structuring this knowledge is not straightforward and requires the involvement of domain experts to ensure the collection of their knowledge and the validation of the generated output to maintain trust and reliability [2]. The proposed system aims at addressing several critical needs: (i) enable the manual encoding of tacit knowledge from domain experts, helping transform experiential know-how into machine-readable data, (ii) facilitate human-in-the-loop validation and augmentation of outputs generated by AI-based extraction tools, ensuring that the resulting knowledge is accurate, and (iii) support the creation of training data for automated knowledge extraction systems, enhancing their learning capabilities through structured, expert-curated input. This paper presents the open source `rapid-triples` tool[1], a novel interface for the collection and structuring of knowledge according to a target ontology for semantic interoperability. Our tool supports human-AI collaboration workflows relying on humans to complete, validate, and enrich (machine-generated) knowledge. Specifically, `rapid-triples` supports both manual and semi-automatic knowledge collection through adaptive forms that can be tailored to the specific use case considered.

The remainder of this paper is organised as follows. Section 2 discusses the human-AI workflows envisioned and enabled by the proposed solution. Section 3 describes `rapid-triples` and its functionalities, while Section 4 exemplifies its usage in the industrial domain to collect procedural knowledge. Section 5 presents related work on interfaces to facilitate the collection of structured knowledge in

HAIBridge 2025: 1st Workshop on Bridging Hybrid (Artificial) Intelligence and the Semantic Web co-located with the 24th International Semantic Web Conference, Nara, Japan

mario.scrocca@cefriel.com (M. Scrocca); alessio.carenini@cefriel.com (A. Carenini); valentina.carriero@cefriel.com (V. A. Carriero); irene.celino@cefriel.com (I. Celino)
0000-0002-8235-7331 (M. Scrocca); 0000-0003-1948-807X (A. Carenini); 0000-0003-1427-3723 (V. A. Carriero); 0000-0001-9962-7193 (I. Celino)

[1] https://github.com/cefriel/rapid-triples

RDF format and compares them with `rapid-triples`. Finally, Section 6 draws the conclusions and suggests future work.

2. Workflows for Human-in-the-loop Knowledge Collection

Constructing high-quality Knowledge Graphs (KGs) aligned to a reference ontology remains a complex and multifaceted challenge. A critical bottleneck lies in the collection and curation of knowledge, which often requires human expertise, contextual understanding, and iterative refinement. In many real-world scenarios, the human actors involved in the process are not experts in ontologies, RDF, or semantic technologies. Therefore, it is essential to support them through intuitive workflows that abstract away technical complexities while still enabling the creation of semantically rich and ontology-compliant knowledge.

To address this problem, we envision a hybrid intelligence approach that combines automated techniques with human-in-the-loop workflows to improve accuracy, completeness, and semantic alignment of the collected knowledge. We identify three representative workflows to be supported:

W1 *Tacit Knowledge Collection*: In this case, the relevant knowledge resides in the minds of domain experts and is not yet documented. The user must be guided in articulating and formalising this tacit knowledge in a structured form that conforms to the target ontology.

W2 *Knowledge Completion from Unstructured Sources*: Here, initial knowledge is automatically extracted from unstructured content (e.g., documents, reports) using AI-based tools. The extracted information serves as a draft that the user must validate, correct, and enrich. This human-in-the-loop process ensures that the resulting knowledge is accurate, complete and semantically consistent with the reference ontology, overcoming possible mistakes of the automatic extraction.

W3 *Enhance Automatic Knowledge Extraction Systems*: The structured knowledge, validated and reviewed by users, is used as training or contextual data to enhance the accuracy of automatic knowledge extraction solutions.

3. rapid-triples: Design and Implementation

To address the challenges of effectively modelling structured knowledge while minimising domain user exposure to RDF, we propose a form-based solution supporting the three knowledge collection workflows identified in Section 2.

Figure 1: `rapid-triples` to implement human-in-the-loop knowledge collection

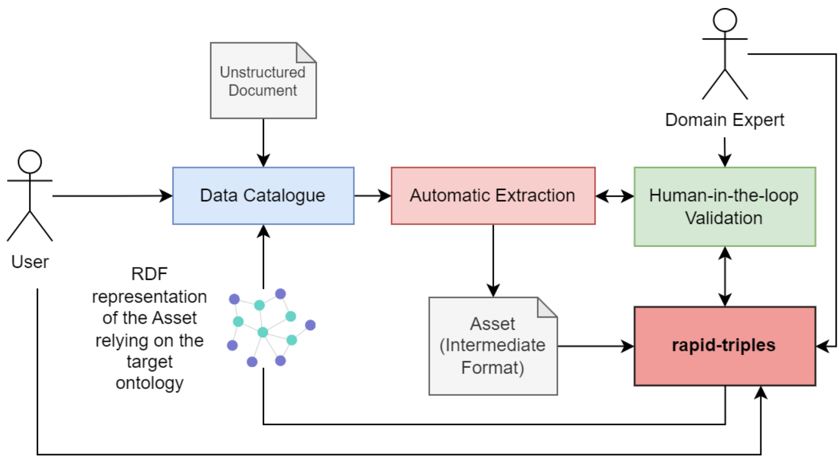
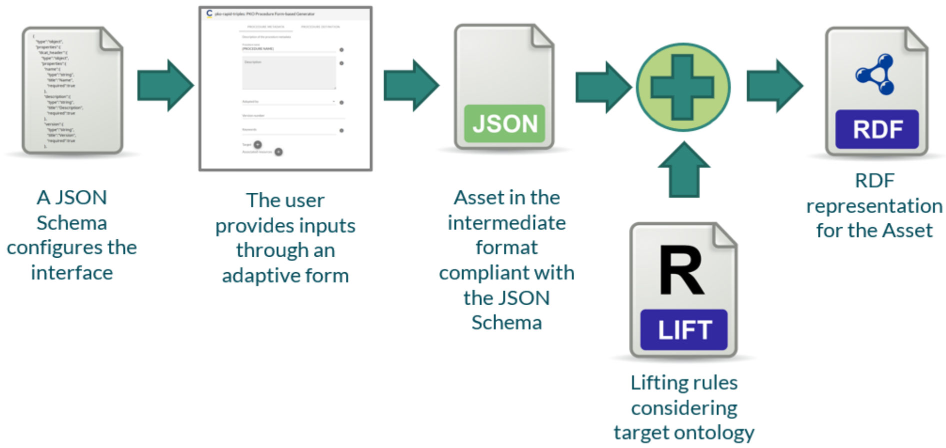


Figure 1 shows the main components envisioned. A *Data Catalogue* represents the point of access for users to structured knowledge represented using a target ontology in a knowledge graph. The `rapid-triples` tool is designed to support: (i) the manual insertion of tacit knowledge via a form-based interface that guides the user and generates a proper RDF output to be added to the knowledge graph (***W1***), (ii) the possibility for domain experts to use the same form to review and update knowledge extracted by *Automatic Extraction* components from existing documents (***W2***), (iii) the provision of expert-curated data collected via the form to the *Automatic Extraction* components to be used as additional training data to improve existing models or as contextual data to implement on-the-fly enhancements to the automatically extracted output. The last two workflows should be mediated by a *Human-in-the-loop Validation* component that manages assignment of tasks to users and handles interactions among components. An *Intermediate Format* representation (i.e., JSON document) is leveraged to facilitate interactions with components that do not support RDF natively.

The `rapid-triples` tool is based on the process in Figure 2 and can be declaratively customised, providing: (i) a compliant JSON Schema [3] that is used to render the form interface and validate the inserted content, and (ii) a set of declarative transformation rules to convert JSON to RDF according to the target reference ontology. The solution hides the classes and properties defined by the target ontology from the users (which is a well-known roadblock to adoption of ontologies in user interaction systems [4, 5]), but it guides them via an interactive interface through the collection of structured knowledge. Despite the origin of the collected knowledge, the tool guarantees its validation, enables its update by the user and is capable of generating a valid RDF output.

Figure 2: Diagram of the underlying process for the `rapid-triples` tool.



The implementation of the `rapid-triples` tool is composed of two main components: a Web interface in Vue.js[2] for creating, editing, and visualising procedures and a corresponding backend component written in Python for storage, processing and interaction with other software components. The frontend component enables an interface for manual insertion by leveraging the `vuetify-json-schema-form`[3] library. The web interface can be used as a standalone component and is made available open source on GitHub. The frontend allows defining a template[4] to generate the target RDF directly client-side, while the backend can execute fully declarative RML mapping rules [6] via the `morph-kgc` processor [7]. The `rapid-triples` tool available on GitHub can be customised by simply modifying the JSON Schema and the template files as specified in the README.

The tool accepts and renders all valid JSON Schemas, but the schema can be enriched with specific directives[5] to personalise the interface, i.e., how the form is rendered to the user. As an example, the

[2]https://vuejs.org/
[3]https://github.com/koumoul-dev/vuetify-jsonschema-form
[4]https://mozilla.github.io/nunjucks/
[5]https://koumoul-dev.github.io/vuetify-jsonschema-form/latest/

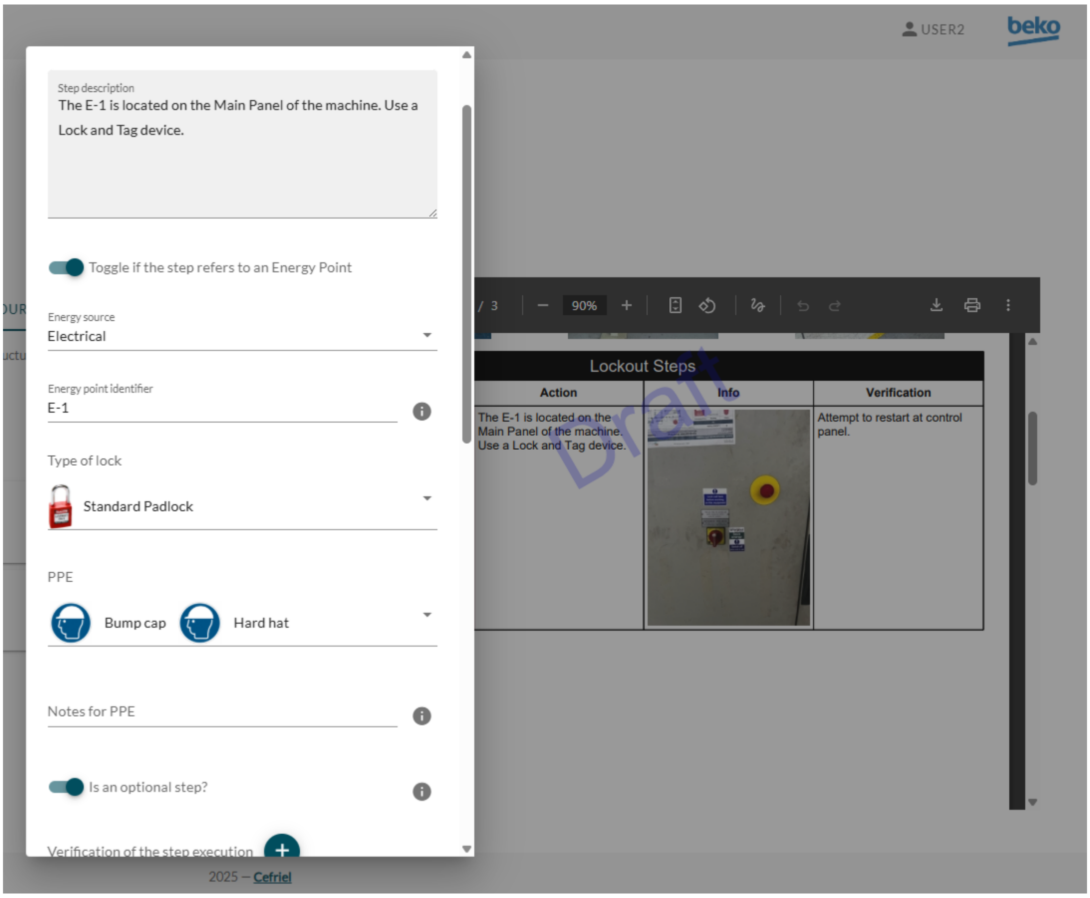
interface can be customised to use specific terminology that is more familiar to the user, but, at the same time, can be mapped properly to the target ontology. Similarly, a field of the form can be rendered using custom visual components (e.g., a dropdown with image selections) or dynamically retrieved values (e.g., from APIs) through special directives. While this is not mandatory, it is preferable to define a JSON Schema aligned to the semantics of the target ontology. If possible for the considered use case, a proper JSON-LD context can be used to convert a JSON compliant with the JSON Schema to the target ontology.

4. rapid-triples for Procedural Knowledge Collection

This section discusses the adoption of the `rapid-triples` tool within the PERKS project[6] for the collection of procedural knowledge relying on the Procedural Knowledge Ontology (PKO) [8]. A publicly available instance of the frontend component for a generic procedure is available online[7].

Within the project, the `rapid-triples` tool has been customised to address the specificities of each use case (e.g., considering custom ontology modules or language) and adapted to enhance the user experience (e.g., developing custom visual components). Moreover, it has been integrated with solutions for AI-based extraction of procedures from documents. The Beko use case in PERKS targets Lock-Out Tag-Out (LOTO) safety procedures, which define a precise list of tasks to be mandatorily executed to ensure that dangerous equipment is properly shut off and not able to be started up again prior to the completion of maintenance or repair work. Figure 3 shows the customisation of the tool for this use case.

Figure 3: Interface for procedural knowledge collection for the Beko use case.



[6]https://perks-project.eu/
[7]https://perks-project.github.io/pko-rapid-triples

The form has been initialised by automatically processing the PDF document shown in the background. The user is now leveraging the tool to further edit and enrich the procedure. While the user sees terms like *energy point* and custom visualisations (e.g., images for the selection of locks and personal protective equipment), the underlying JSON Schema and the declarative mapping rules guarantee an interoperable output using the classes and properties from PKO (e.g., `pko:Procedure`) and from the custom ontology module defined for Beko (e.g., `pko-beko:MachineEnergyPoint`).

Within the PERKS project, we run a preliminary qualitative evaluation of the `rapid-triples` tool with Beko users, testing both the manual knowledge collection and the completion of automatically extracted procedures. The users were able to generate a valid RDF representation of LOTO procedures using PKO for different machines in the factory, and we then compared the existing LOTO procedures with the ones collected via the `rapid-triples` tool. The domain experts appreciated the manual collection through the adaptive form, as it effectively guided them in documenting additional tacit knowledge while providing a better user experience with respect to paper or tabular-based approaches. Moreover, they positively evaluated the implemented workflow for the human-in-the-loop collection of procedural knowledge from existing documents. In particular, they liked the seamless experience compared to the manual approach. Finally, safety managers highlighted the higher quality of the generated procedures, and LOTO operators valued the possibility of having access to more details during the procedure execution.

5. Related Work

The challenge of enabling high-quality RDF authoring by users less familiar with Semantic Web standards has spurred the development of numerous tools that balance customizability and reusability of the solutions. One common strategy is the use of schema-aware assistance to dynamically customise the interfaces based, for example, on the target ontology or the definition of SHACL shapes. *ActiveRaUL* takes a web form-based approach, automatically generating RDF-editing interfaces from ontologies to separate schema modelling from data creation [9]. *Schímatos* [10] is a web application utilising the Shapes Constraint Language (SHACL) to define user-friendly interfaces for generating and modifying forms based on shapes. The shapes define the interface and are used to validate the user inputs thus minimising errors. RDForms [11, 12] is a Javascript library to configure HTML forms for editing and updating RDF, but is based on a custom language for configuring the forms and not on a standard language.

The proposed `rapid-triples` tool also employs a form-based approach but leverages a JSON Schema definition to automatically instantiate the interface and validate the content. Unlike the other tools mentioned, we chose to decouple the serialization of the form content from the target RDF representation. Indeed, the JSON intermediate form facilitates the integration of `rapid-triples` with other AI-based solutions and human-in-the-loop interfaces not capable of directly processing RDF. Declarative mapping rules for knowledge graph construction based on RML [6] or template-based approaches [13] are supported to map the JSON Schema used to configure the form to a valid and consistent output in RDF. An additional advantage of the proposed decoupled approach is related to cases in which a portion of the expected RDF output can be determined without requiring an explicit input. In these scenarios, the `rapid-triples` tool can be configured to collect only the minimum required information from users, thereby reducing the overall effort. We identified two main cases: (i) fixed characteristics, i.e., attributes or relations that are consistently present across all instances of a given asset type, and (ii) parametric characteristics, i.e., entities with a predefined structure that must be instantiated in similar ways depending on a specific value. Fixed characteristics can be automatically initialised for a given entity type without a direct correspondence in the form structure. Similarly, parametric entities can request from users only the necessary values. The instantiation of the underlying RDF triple patterns is handled automatically within the mapping rules.

The *FORMULIS* system [14, 15] has introduced dynamic and nested form interfaces to bridge usability and expressivity in RDF authoring. The initial design focused on guiding users through RDF creation by adapting form fields and values based on previously entered content. This approach was refined further to include nested forms and intelligent filling suggestions, helping users produce semantically rich and high-quality RDF while abstracting away the RDF syntax. *Shacled Turtle* offers a method for Turtle auto-completion using RDFS and SHACL schemas, improving the authoring experience for users familiar with RDF syntax but in need of structured guidance [16]. Similarly, the `rapid-triples` tool provides mechanisms to enable adaptation of the form based on the user's input and dynamic retrieval from external data sources (e.g., APIs). The main difference is that while *FORMULIS* and *Shacled Turtle* leverage directly the target RDF vocabularies to provide suggestions, the `rapid-triples` tool uses the JSON Schema.

Other approaches for the collection of structured knowledge in RDF are based on the annotation of existing documents. OntoPawls [17] is an interface to annotate specific portions of a PDF document according to the classes and properties of a given ontology. However, expecting users to annotate documents directly is often overly demanding, so we identified the need to design a tool that guides users in the process and integrates with automatic methods for knowledge extraction from existing data sources.

6. Conclusions

We presented the `rapid-triples` tool, a flexible form-based interface for domain users to perform knowledge graph construction and editing. The form can be configured via a JSON Schema and easily customised to address specific requirements. The intermediate JSON representation handled by the tool facilitates the integration of AI-based solutions for human-in-the-loop knowledge collection and enables a coherent validation and lifting process to the RDF representation according to the target ontology. The frontend component of the `rapid-triples` tool is available open source and can be used as a standalone solution or integrated with AI-based solutions to implement complex workflows through a dedicated backend component. The tool has been preliminarily validated in the PERKS project, considering the collection of procedural knowledge in industrial settings, and a new development and evaluation phase is planned in the next months.

The future steps will focus on refining the designed interface and further smoothing the user experience in order to reduce the perceived overload [17]. Additionally, we aim to further explore and investigate different human-in-the-loop interaction patterns (e.g., extending those introduced in [18, 19]), thus exploiting the combination of the tool with various AI-based solutions. Finally, considering the lifting mapping rules, we plan to investigate solutions combining declarative mapping rules and automatic methods to handle parametric procedures that may benefit from the natural language generation capabilities of language models in the KG construction process [20].

Acknowledgments

This work is partially supported by the PERKS project, co-funded by the European Commission (Grant id 101120323).

Declaration on Generative AI

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

References

[1] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, X. Wu, Unifying large language models and knowledge graphs: A roadmap, IEEE Trans. on Knowl. and Data Eng. 36 (2024) 3580–3599. URL: https://doi.org/10.1109/TKDE.2024.3352100. doi:10.1109/TKDE.2024.3352100.

[2] I. Celino, V. A. Carriero, A. Azzini, I. Baroni, M. Scrocca, Procedural knowledge management in Industry 5.0: Challenges and opportunities for knowledge graphs, Journal of Web Semantics 84 (2025) 100850. URL: https://www.sciencedirect.com/science/article/pii/S1570826824000362. doi:https://doi.org/10.1016/j.websem.2024.100850.

[3] F. Pezoa, J. L. Reutter, F. Suarez, M. Ugarte, D. Vrgoč, Foundations of json schema, in: Proceedings of the 25th international conference on World Wide Web, 2016, pp. 263–273.

[4] G. Aguado de Cea, E. Montiel Ponsoda, M. C. Suárez-Figueroa, Approaches to ontology development by non ontology experts, in: Proceedings of International Symposium on Data and Sense Mining, Machine Translation and Controlled Languages, ISMTCL, Presses universitaires de Franche-Comté, Besanon, Francia, 2009.

[5] J. Attard, F. Orlandi, S. Auer, Exconquer: Lowering barriers to rdf and linked data re-use, Semantic Web 9 (2018) 241–255.

[6] D. Van Assche, T. Delva, G. Haesendonck, P. Heyvaert, B. De Meester, A. Dimou, Declarative RDF graph generation from heterogeneous (semi-)structured data: A systematic literature review, Web Semant. 75 (2023). doi:10.1016/j.websem.2022.100753.

[7] J. Arenas-Guerrero, D. Chaves-Fraga, J. Toledo, M. S. Pérez, O. Corcho, Morph-KGC: Scalable knowledge graph materialization with mapping partitions, Semantic Web (2022) 1–20. doi:10.3233/SW-223135.

[8] V. A. Carriero, M. Scrocca, I. Baroni, A. Azzini, I. Celino, Procedural knowledge ontology (pko), in: The Semantic Web: 22nd European Semantic Web Conference, ESWC 2025, Portoroz, Slovenia, June 1–5, 2025, Proceedings, Part II, Springer-Verlag, Berlin, Heidelberg, 2025, p. 334–350. URL: https://doi.org/10.1007/978-3-031-94578-6_19. doi:10.1007/978-3-031-94578-6_19.

[9] A. S. Butt, A. Haller, S. Liu, L. Xie, ActiveRaUL: A Web form-based User Interface to Create and Maintain RDF data, in: Proceedings of the ISWC 2013 Posters & Demonstrations Track within the 12th International Semantic Web Conference (ISWC 2013), 2013. URL: https://ceur-ws.org/Vol-1035/iswc2013_demo_30.pdf.

[10] J. Wright, S. J. Rodríguez Méndez, A. Haller, K. Taylor, P. G. Omran, Schímatos: A shacl-based web-form generator for knowledge graph editing, in: The Semantic Web – ISWC 2020: 19th International Semantic Web Conference, Athens, Greece, November 2–6, 2020, Proceedings, Part II, Springer-Verlag, Berlin, Heidelberg, 2020, p. 65–80. URL: https://doi.org/10.1007/978-3-030-62466-8_5. doi:10.1007/978-3-030-62466-8_5.

[11] M. Palmér, F. Enoksson, M. Nilsson, A. Naeve, Annotation profiles: Configuring forms to edit rdf, in: Proceedings of the International Conference on Dublin Core and Metadata Applications, Dublin Core Metadata Initiative, 2007.

[12] MetaSolutions, RDForms - RDF in HTML-forms, 2009. URL: https://rdforms.org.

[13] M. Scrocca, A. Carenini, M. Grassi, M. Comerio, I. Celino, Not everybody speaks RDF: Knowledge conversion between different data representations, in: Fifth International Workshop on Knowledge Graph Construction@ ESWC2024, 2024. URL: https://ceur-ws.org/Vol-3718/paper3.pdf.

[14] P. Maillot, S. Ferré, P. Cellier, M. Ducassé, F. Partouche, FORMULIS: Dynamic Form-Based Interface for Guided Knowledge Graph Authoring, Springer International Publishing, 2017, pp. 140–144. URL: http://dx.doi.org/10.1007/978-3-319-58694-6_18. doi:10.1007/978-3-319-58694-6_18.

[15] P. Maillot, S. Ferré, P. Cellier, M. Ducasse, F. Partouche, Nested forms with dynamic suggestions for quality RDF authoring, 2017, pp. 35–45. doi:10.1007/978-3-319-64468-4_3.

[16] J. Bruyat, P.-A. Champin, L. Médini, F. Laforest, Shacled turtle: Schema-based turtle auto-completion, Workshop on Visualization and Interaction for Ontologies and Linked Data 2022, co-located with the International Semantic Web Conference 2022 (2022). URL: https://ceur-ws.org/Vol-3253/paper1.pdf.

[17] A. Rula, G. Re Calegari, A. Azzini, I. Baroni, I. Celino, Annotation and extraction of industrial procedural knowledge from textual documents, in: Proceedings of the 12th Knowledge Capture Conference 2023, 2023, pp. 1–8.
[18] H. F. Witschel, C. Pande, A. Martin, E. Laurenzi, K. Hinkelmann, Visualization of Patterns for Hybrid Learning and Reasoning with Human Involvement, Springer International Publishing, Cham, 2021, pp. 193–204. URL: https://doi.org/10.1007/978-3-030-48332-6_13. doi:10.1007/978-3-030-48332-6_13.
[19] S. Tsaneva, D. Dessì, F. Osborne, M. Sabou, Knowledge graph validation by integrating LLMs and human-in-the-loop, Information Processing & Management 62 (2025) 104145. URL: https://www.sciencedirect.com/science/article/pii/S030645732500086X. doi:https://doi.org/10.1016/j.ipm.2025.104145.
[20] X. Liang, Z. Wang, M. Li, Z. Yan, A survey of llm-augmented knowledge graph construction and application in complex product design, Procedia CIRP 128 (2024) 870–875. URL: https://www.sciencedirect.com/science/article/pii/S2212827124007911. doi:https://doi.org/10.1016/j.procir.2024.07.069, 34th CIRP Design Conference.