PERKS: Eliciting and Exploiting Procedural Knowledge in Industry 5.0

Ioan Toma[1], Umutcan Serles[1,2]

[1]Onlim GmbH, Weintraubengasse 22, A-1020 Wien, Austria
[2]University of Innsbruck, Technikerstrasse 21a, A-6020 Innsbruck, Austria

Abstract

Procedural Knowledge (PK) is essential for industrial operations but often challenging to manage. The PERKS project addresses this by developing AI-based tools and methodologies to capture, manage, and deliver PK throughout its lifecycle. By focusing on human-centric solutions and testing in diverse industrial settings, PERKS aims to improve operational efficiency and worker performance while enabling knowledge transfer across industries. The PERKS results are applied and tested in three industrial scenarios (white goods production plant, computer numerical control machines, microgrid testbed) providing different use cases in terms of PK complexity and industrial requirements.

Keywords

Procedural Knowledge, Knowledge Graphs, LLMs, Hybrid AI

1. Introduction

PERKS "Eliciting and Exploiting Procedural Knowledge in Industry 5.0"' is a Horizon Europe collaborative research project funded by the EU. It started in October 2023 and is expected to last 36 months. The PERKS project supports the holistic governance of industrial Procedural Knowledge (PK) in its entire life cycle, from elicitation to management and from access to exploitation [1]. PERKS bases its solutions on leading-edge AI (both symbolic and sub-symbolic) and data technologies, by advancing and integrating existing methodologies and tools in terms of readiness, flexibility and user acceptance. Besides AI and data, the third pillar of PERKS consists of people: the goal is to put industry workers at the centre, in line with the Industry 5.0 vision, to satisfy their concrete needs, to provide AI-powered digital tools to perform their tasks better and more easily, following a human-in-the-loop paradigm to enhance the technologies and the solutions. This paper presents the PERKS research project[1] which aims to provide the support for eliciting and exploiting procedural knowledge in Industry 5.0. The reminder of this paper is organized as follows. Section 2 describes the main objectives of the project, Section 3 presents the project's current status and Section 4 introduces the PERKS solution and details the current implementation of one of the PERKS use cases. Finally Section 5 reports on potential interests and collaborations with other projects.

2. Objectives

PERKS objectives are standardizing the digital representation of procedures, investigating the best methods and technologies to elicit procedures from documents and people experience, designing solutions based on knowledge graphs [2] to manage, preserve and give access to procedures, and providing tools and systems to allow people to search and find procedural knowledge when and where they need it. Moreover, PERKS aims for a holistic governance throughout the entire life cycle, because knowledge evolves and changes over time. The rest of this section details the objectives of the PERKS project:

RuleML+RR'24: Companion Proceedings of the 8th International Joint Conference on Rules and Reasoning, September 16–22, 2024, Bucharest, Romania

ioan.toma@onlim.com (I. Toma); umutcan.serles@onlim.com (U. Serles)

[1]https://perks-project.eu/

1. **AI-powered tools to assist industrial workers in managing and executing procedures**. These tools will be designed to improve efficiency, productivity, and knowledge accessibility. The tools effectiveness will be measured through usability assessments, performance metrics, and data transparency evaluations. The developed tools will be integrated into real-world industrial settings to assess their impact.
2. **Standardized ontology for representing procedural knowledge.** This ontology will facilitate knowledge sharing and reuse across different industrial contexts. By aligning with existing standards and ensuring interoperability, the ontology will enable the development of AI-powered tools and services. PERKS will evaluate the ontology's effectiveness through rigorous testing and validation in real-world industrial scenarios.
3. **Improving AI and data technologies through human-in-the-loop collaboration**. By involving industry experts, PERKS aims to enhance the reliability, transparency, and accountability of AI systems. This approach will be measured by evaluating system performance, user satisfaction, and alignment with trust and safety standards.
4. **Practical impact of the project's solutions in real-world industrial settings**. By applying the developed tools and methodologies to diverse use cases, PERKS aims to measure improvements in productivity, efficiency, and sustainability. The project will utilize established metrics and frameworks to assess economic, social, and environmental impacts.

3. Current Project Status

The PERKS digital solution makes use of Artificial Intelligence to better manage the set of skills, information, and experiences related to "know-how" in industrial processes. The project started 10 months ago and involves a consortium of technical partners (Cefriel, DFKI, NTT Data, Onlim, Vienna University of Economics and Business) as well as industrial use case companies (BEKO Europe, Fagor Automation and Siemens). In these first months, the extensive discussions and collaborative workshops between the partners led to a set of clear and shared business and technical requirements. Those represent a blueprint for a large set of corporate scenarios beyond the boundaries of the project scope, as also testified by the initial market analysis carried out by International Data Corporation (IDC). The envisioned solution will represent procedural knowledge in a form that AI applications can easily leverage to support different stakeholders to manage, execute, audit and validate both the procedure definitions as well as their application on the field. In the next period, the PERKS project will build upon those foundations and will implement a first version of the digital solution to be deployed and tested in BEKO Europe, Fagor Automation and Siemens, to prove the effectiveness of the proposed approach to realize the PERKS vision.

4. PERKS Solution

PERKS solution consists of various modules that enable AI-supported creation and consumption of procedural knowledge. In the following, we will first briefly introduce each module and give an indicative example of how a conversational agent can assist a user with the consumption of procedural knowledge.

4.1. Modules

Figure 1 shows an overview of the PERKS solution [3]. The solution consists of 6 major modules:

- **Knowledge Extraction and Collection:** The main responsibility of this module is to generate procedural knowledge from various sources with different nature with different levels of automation. This includes automated extraction of procedural knowledge from unstructured text (e.g., PDF files), semi-automated generation from semi-structured data (e.g., JSON, XML, HTML files) and manual creation of procedural knowledge by domain experts.

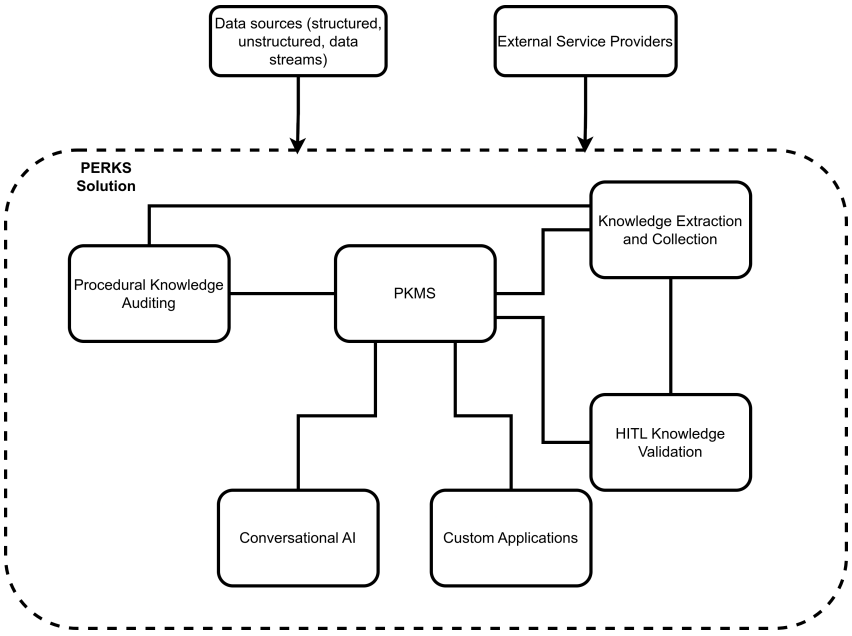

Figure 1: An overview of the PERKS solution and its components.

- **Human-in-the-loop (HITL) knowledge validation:** This module enables domain experts to validate and approve generated knowledge [4]. This module is particularly important to ensure the correctness of (semi-)automatically generated knowledge.
- **Procedural knowledge auditing:** This module provides a mechanism for domain experts to audit executed procedures to identify pitfalls and diagnose problems with the execution.
- **Procedural Knowledge Management System (PKMS):** PKMS is the core module of the PERKS solution. It manages the lifecycle of procedural knowledge graph [5], including formal verification and hosting of created knowledge in a knowledge graph, versioning and querying of the procedures in the knowledge graph.
- **Conversational AI:** This module provides the main interaction mechanism with the procedural knowledge for the end-users (e.g., operators in a factory). The conversations can be structured (e.g., state-based) based on the procedural knowledge stored in the knowledge graph, but can be also in a question answering setting where the questions are answered over unstructured text, including automatically generated FAQs.
- **Custom Applications:** A toolbox of applications specifically designed to meet the unique requirements of a certain use case.

4.2. Indicative Prototype: Beko Europe Use Case

Beko Europe manufactures home appliances in large factories. The manufacturing process is involved with many procedures, including LOTO (Lockout TagOut) procedures, which are series of steps that needs to be applied to take equipment used in manufacturing out of service in order to enable safe maintenance and servicing. LOTO procedures are typically more than just turning off a switch and needs to be carried by authorized personnel in a systematic manner.

We prototyped an LLM-based conversational agent that can make use of the procedural knowledge stored in a knowledge graph with the LangChain framework[2]. This prototype only implements the conversational AI module supported by a primitive representation of a procedure as a schema:HowTo instance[3]. The agent uses the ReAct prompting technique[6], which allows an LLM agent to mimic human reasoning process and take certain actions based on thoughts and observations.

[2]https://www.langchain.com/
[3]See the full schema here: https://semantify.it/ds/eh3AMgToN

Listing 1: The zero-shot ReAct prompt for recursive procedure consumption

```
1  You are a bot that presents procedural knowledge from a knowledge base. Complete the objective as best you can. You have
       access to the following tools for you to access to the knowledge base:
2
3  {tools}
4
5  Use the following format:
6
7  Question: The input question you must answer
8  Thought: You should always think about what to do.
9  Action: the action to take, should be one of [{tool_names}]
10 Action Input: the input to the action
11 Observation: the result of the action
12 Thought: you need to think about what do with the Observation
13 ... (this Thought/Action/Action Input/Observation sequence can repeat multiple times)
14 Thought: I now know the final answer
15 Final Answer: the final answer to the original input question
16
17 Here are some instructions for you:
18
19 If the question asks for a procedure, then look for the procedure with the GetProcedure tool in your action. The action input
       is the name of the procedure.
20
21 start_function
22 The observation O will tell you a procedure consisting of zero or more steps. A step in O is in the form of "{{<uri>}}_{{
       position}}_{{step text}}".
23
24 For each Step in O
25
26 - Produce a thought about searching for the substeps of Step. Produce an action GetProcedure. Produce Action Input in the
       form of "{{uri}}_{{step text}}".
27 - The user will give you a new Observation O1. A step in O1 is in the form of "{{<uri>}}_{{position}}_{{step text}}".
28 - If O1 is "There is no substep.", produce an action and action input with the next Step in O
29 - If O1 is something else, apply the instructions between start_function and end_function to O1.
30
31 - Produce a thought about knowing the final answer when the substeps of all steps and their substeps are checked recursively
       with the loop above and observation for each produced "There is no substep.".
32 end_function
33
34 Apply the instructions between start_function and end_function to the question.
35
36 Present Final Answer by using the information in each Observation. Order the steps by position.
37 Begin!
38
39 Question: {input}
```

Listing 1 shows the prompt used for the prototype. Line 1 is an introductory prompt to tell the agent its task. Line 3 is a place holder for available tools. These typically functions that generate observations for the agent based on some external source. In our case, we use only one tool called GetProcedure, which runs a single SPARQL query with some given parameters to retrieve the (sub)steps of a procedure. Lines 7-15 presents the expected output from the agent in an abstract way. Given a question, the agent is expected to produce a series of thought-action-observation patterns until it reaches to final answer. The lines 17-32 define a recursive function including an exit condition in natural language. This way, the agent knows until when it should produce the thought-action-observation patterns. The lines 34-36 calls the defined function and provide some formatting instructions. Line 39 is a place holder for the input.

Listing 2: An excerpt from the ReAct prompting outcome for querying a LOTO procedure

```
Thought: I need to retrieve the procedure for turning off pneumatics.
Action: GetProcedure
Action Input: pneumatics

Observation:<https://onlim.com/entity/Nec0fa475954d492abb722adabcde63d1>_1- Disconnection Process <https://onlim.com/entity/
    N8684a77c332a40bfab4909f8189fff32>_2- Verification
I now have the first step of the procedure, which is the disconnection process. Now I need to search for the substeps of this
     step.
Action: GetProcedure
Action Input: <https://onlim.com/entity/Nec0fa475954d492abb722adabcde63d1>_Disconnection Process

Observation:<https://onlim.com/entity/Ne2e04fd142fe45749b53d867c7e5c3f7>_1- Rotate ball valve A-1 90 degrees from open (
    vertical) to closed (horizontal) position
...
I have found the substeps for the disconnection process. Now I need to search for the substeps of each substep.
Action: GetProcedure
Action Input: <https://onlim.com/entity/Ne2e04fd142fe45749b53d867c7e5c3f7>_Rotate ball valve A-1 90 degrees from open (
    vertical) to closed (horizontal) position

....

Observation:There is no substep.
Thought: I now know the final answer.
Final Answer:
1. Disconnection Process:
   - Rotate ball valve A-1 90 degrees from open (vertical) to closed (horizontal) position
   - Apply ed block the red padlock ed the red TAG
   - Discharge the residual air present in the circuit using the gun connected to the T under valve A-1
2. Verification:
```

```
- Check that no more air comes out of the gun and that the pressure on the pressure gauges on the machine has dropped to zero
```

Listing 2 shows an excerpt from a run of the agent with the input ”turn off pneumatics”. The agent creates an initial thought about retrieving the procedure for turning of the pneumatics. Following this thought, it takes an action to call the GetProcedure function. GetProcedure function parses the input and uses the name as a parameter to the SPARQL query to retrieve the steps of procedure that contains the name ”pneumatics”. The result of this function produces an observation. An observation contains the URI of the returned step, its position and its name. Due to its instructions, the agent produces thoughts actions and observations for each returned procedure step recursively, until the SPARQL query returns no results, leading to a depth-first traversal of the procedure. Finally, it presents the answer, the entire solution.

The advantage of this approach is that logic of the agent is mostly in the prompt, which gives the developers flexibility. The recursion logic does not have to be hard-coded in the application. Only the GetProcedure function needs to be implemented to call a single SPARQL query whenever a new procedure step is needed. The LLM agent decides when the procedure retrieval process is over.

5. Collaboration with other projects

PERKS actively seeks collaboration with projects operating in similar or complementary industrial domains to enrich its procedural knowledge ontology and refine its technological solutions. By engaging with partners from diverse sectors, the project aims to identify additional use cases and requirements that can be incorporated into the ontology’s structure and functionality. This collaborative approach will enhance the ontology’s adaptability and applicability across various industrial contexts, ultimately strengthening the project’s outcomes. Furthermore, collaboration with other projects offers the opportunity to leverage existing tools and methodologies, avoiding redundant efforts and accelerating development. By sharing knowledge and resources, PERKS can build upon the strengths of partner projects to create more robust and comprehensive solutions.

Acknowledgments

This work is funded by the HORIZON Innovation Action 101120323 PERKS “Eliciting and Exploiting Procedural Knowledge in Industry 5.0”.

References

[1] A. Rula, G. R. Calegari, A. Azzini, D. Bucci, I. Baroni, I. Celino, Eliciting and curating procedural knowledge in industry: Challenges and opportunities, in: Proceedings of the Third Conference on Digital Curation Technologies (Qurator 2022), Berlin, Germany, Sept. 19th-23rd, 2022, volume 3234, 2022.

[2] D. Fensel, U. Simsek, K. Angele, E. Huaman, E. Kärle, O. Panasiuk, I. Toma, J. Umbrich, A. Wahler, Knowledge graphs: Methodology, tools and selected use cases, Knowledge Graphs (2020). URL: https://api.semanticscholar.org/CorpusID:210975360.

[3] M. Hulea, M. Scrocca, V. Carriero, N. Pokaratsiri, S. Ostermann, U. Serles, I. Toma, J. Xavier Parreira, E. Flitz, L. Waltersdorfer, et al., Deliverable D4.1 Reference Architecture for a Procedural Knowledge Management System, Technical Report, PERK Project, 2024.

[4] S. Tsaneva, M. Sabou, Enhancing human-in-the-loop ontology curation results through task design, ACM J. Data Inf. Qual. 16 (2024) 4:1–4:25.

[5] U. Simsek, K. Angele, E. Kärle, J. Opdenplatz, D. Sommer, J. Umbrich, D. Fensel, Knowledge graph lifecycle: Building and maintaining knowledge graphs., in: KGCW@ ESWC, 2021.

[6] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, Y. Cao, React: Synergizing reasoning and acting in language models, arXiv preprint arXiv:2210.03629 (2022).