Towards an Ontology for Procedural Knowledge in Industry 5.0*

Valentina Anita Carriero, Mario Scrocca, Ilaria Baroni, Antonia Azzini and Irene Celino

Cefriel – Politecnico di Milano, Milan, Italy

Abstract

Procedural Knowledge (PK) refers to the knowledge required to perform specific tasks. In the industrial domain, PK refers to structured processes to be followed, e.g., on the production line of a plant. Oftentimes, such knowledge is not explicitly documented, or, when documented, the only digital source of PK is in an unstructured format, thus it is difficult to access, retrieve and exploit. AI and data-driven tools can support the holistic governance of industrial PK in its entire life cycle, from elicitation to management, from access to exploitation. This paper describes the process of requirements collection for PK management, considering three heterogeneous use cases, and presents the preliminary results obtained. We identify and discuss six main conceptual areas for the PK domain, each including different concepts that should be considered in the modelling of different PK aspects. Such analysis drives the development of an ontology that will enable a modular architecture for a Procedural Knowledge Management System (PKMS) relying on a shared and interoperable representation of PK.

Keywords

industry 5.0, procedural knowledge, ontology, knowledge engineering

1. Introduction

Procedural Knowledge (PK) is *knowing how* to perform some tasks, as opposed to descriptive/declarative knowledge, which is *knowing what* in terms of facts and notions. In industry, PK refers in general to structured processes to be followed, and can be related to both production (e.g., procedure on the production line in a plant) and services (e.g., procedure for troubleshooting during customer support); to specific technical expertise (e.g., procedure to set up a specific machine) and general regulations and best practices (e.g., safety procedures, activities to minimise environmental impact).

Process mining techniques (e.g., [1, 2, 3]) can offer a solution to address PK management, making it possible to analyse and optimise industrial operations. However, this becomes a challenge, especially in industries that are still in the early stages of their digital transformation: (a portion of) their procedural knowledge is not explicitly documented and includes common sense, which usually remains tacit. Even when documented, the only digital source of PK is in an unstructured format, expressed by means of natural language, across heterogeneous documents and systems. The lack of explicit PK and the difficulty to access and retrieve PK for those who need to apply it, result in a partial/poor compliance with industrial processes and standards, specifically: (i) lack of clear procedures shared between operators, leading to heterogeneity of task execution and results; (ii) undesired errors when executing the procedures, negatively impacting on business objectives, the functioning of industrial services and systems, costs or employees' safety; (iii) difficulty in knowledge transfer and (re)training/onboarding of (new) employees.

Our research&development goal[1] is to support the holistic governance of industrial procedural knowledge in its entire life cycle, from elicitation to management, from access to exploitation of explicit PK, by leveraging advanced AI and data technologies and tools. The overall goal is to optimise industrial

SemIIM 2024: Third International Workshop on Semantic Industrial Information Modelling, co-located with the International Semantic Web Conference (ISWC 2024) - Nov 11, 2024 - Nov 15, 2024, Baltimore, USA

valentina.carriero@cefriel.com (V. A. Carriero); mario.scrocca@cefriel.com (M. Scrocca); ilaria.baroni@cefriel.com (I. Baroni); antonia.azzini@cefriel.com (A. Azzini); irene.celino@cefriel.com (I. Celino)

0000-0003-1427-3723 (V. A. Carriero); 0000-0002-8235-7331 (M. Scrocca); 0000-0001-5791-8427 (I. Baroni); 0000-0002-9066-1229 (A. Azzini); 0000-0001-9962-7193 (I. Celino)

[1] Supported by the PERKS project https://perks-project.eu/

processes in a wide range of scenarios where procedural knowledge plays a key role. The three main pillars of our work are:

- Artificial Intelligence, both symbolic and subsymbolic: we aim to integrate and complement knowledge-based and machine learning approaches in order to provide a semi-automatic support to collect, formalize and exploit procedural knowledge in an explicit and structured form;
- data and data technologies, including FAIR principles and data integration of multiple data sources, such as data from IoT and connected machines;
- people, which complement automatic processing in collecting and validating knowledge, and whose interests are put at the center, since we want to develop tools and technologies that adapt the industrial and operational processes to the need of the workers (as one of the main goals of the Industry 5.0 approach).

To achieve our goal, we aim (1) to define a reference modular architecture for PK management, (2) to realise a set of interoperable and complementary digital tools that can be composed, integrated and customised as use-case specific solutions, based on the specific industrial contexts, and (3) to build a reference conceptual model to ensure a structured and uniform representation of the procedural knowledge, as well as interoperability among the tools across different use cases. This paper presents the ongoing process of requirements collection and analysis, which is a prerequisite to the definition of the reference conceptual model, implemented as an ontology (or a network of interconnected ontologies).

Section 2 presents three concrete industrial use cases from which we collected the ontology requirements. In Section 3, after briefly describing the adopted ontology engineering methodology, we describe the requirements collection activity, and discuss our preliminary analysis of the PK domain, based on the requirements collaboratively refined with domain experts involved in the considered use cases. This activity resulted in the identification of a number of concepts to be modelled, split into six main conceptual areas; such analysis will drive the development of the final ontology. Section 4 concludes the paper.

2. Use Cases

We draw industrial requirements from diverse and representative use cases of different complexity, ensuring a broad validation of its results, and their transferability across contexts and sectors. This section provides an overview of the use cases we considered so far, each representing a real-world scenario where procedural knowledge is key. Moreover, in line with the Industry 5.0 concept, we chose those use cases because they specifically focus on human-centered scenarios, in which digital tools and technologies are not aimed at automating or replacing people's work, but are leveraged to augment employees' capabilities and to ease their workflows.

2.1. Use case 1: replicable procedures

The first use case[2] focuses on enhancing safety during shop floor maintenance by applying the Lockout/-Tagout (LOTO) procedure. A LOTO procedure defines the necessary steps to safely shut off dangerous equipment every time a maintenance activity should be performed. There are two application scenarios: one involves shutting off the entire workstation and heavily affects the operations due to the time required to perform all the steps and the complete cessation of workstation activities; the other allows for specific areas to be isolated based on maintenance needs, however this approach increases procedural complexity. The goal is to support the technicians in the definition and application of LOTO procedures, facilitating access to the correct LOTO workflows and enabling their step-by-step execution. Each factory adopts a different template for LOTO procedures and potentially a different language for their specification based on the factory location. Moreover, many instructions are implicitly assumed and not specified within the procedure, making it difficult, especially for junior technicians, to correctly

[2] Requirements collected from Beko Europe.

and safely execute the procedure. The digitalisation of LOTO procedures could also further support their execution by facilitating access to related resources, e.g., images of specific parts of a workstation involved in a certain step. Specific requirements for this use case are associated with the need for managing different states for a procedure, e.g., to keep track of which procedures have been approved for execution, and the expertise levels of technicians executing a procedure, e.g., to provide a more detailed description of the procedure to junior technicians.

2.2. Use case 2: parametric procedures

The second use case[3] revolves around configuring Computer Numerical Controls (CNCs) systems for machine automation. CNCs require precise parametrization during the setup process, which demands considerable time and specialized skills with each machine. Currently, this invaluable knowledge is transferred verbally between technicians, with basic guidelines in the CNC installation manual but lacking a standardized method for capturing detailed machine-specific information. The objective is to enable the transfer of knowledge related to the machine commissioning process from senior technicians to new junior technicians. Digital tools should support new technicians in executing the machine commissioning process by guiding them in the sequential configuration of the different parameters and by providing support in the retrieval of the required values. Specific requirements for this use case are related to keeping track not only of the different steps executed by the technicians, but also of the errors encountered and the doubts not specifically addressed by the procedure.

2.3. Use case 3: human-machine procedures

The third use case[4] focuses on a Microgrid. This setup connects offices and factories to renewable energy sources, energy storage systems, electric vehicle charging stations, and the external power grid, all managed by a microgrid controller. The system, equipped with sensors, collects real-time data on electricity flow, which is then stored for analysis. Currently, energy optimization is automated but lacks clear documentation and user involvement. The aim is to extract procedures that describe the interactions between user behaviour and the Microgrid components by leveraging expert knowledge and historical data analysis. The identification of such procedural knowledge and the analysis of data described through such procedures could be used to enhance energy management, optimize battery usage, and boost sustainability. Specific requirements for this use case are associated with the need to define and manage procedures that involve not only users but also software components to observe and act on the Microgrid components.

3. PERKS Conceptual Model

The reference modular architecture for PK management (Procedural Knowledge Management System, PKMS) that we are developing is enabled by the definition of a reference conceptual model, in order to ensure a structured and uniform representation of the procedural knowledge within a PKMS, and the interoperability among its components [4]. As we envision a PKMS based on knowledge graphs (KGs) for data management and storage, the reference conceptual model will be formalised as OWL ontologies, defining all the relevant classes and properties to be populated in the KG. Having a shared ontology for representing all data behind such KGs, makes their semantics explicit, enables reasoning and inference, and supports procedural knowledge (PK) reuse and interoperability.

The main requirement to be fulfilled by the conceptual model is to enable a flexible abstraction that could support the representation of procedures according to the specific needs of different use cases. Such model may be extended within a specific implementation of the PKMS to include specific domain-based ontology entities targeting a certain use case, and/or business-related vocabularies and taxonomies that are defined and reused within a company.

[3] Requirements collected from Fagor Automation.

[4] Requirements collected from Siemens, related to their company campus in Vienna.

For developing our procedural knowledge ontology, we rely on the Linked Open Terms (LOT) methodology[5] [5], an industrial method for developing ontologies. The LOT methodology defines iterations over a workflow composed of four main activities: (i) ontological requirements specification, (ii) ontology implementation, (iii) ontology publication, and (iv) ontology maintenance. We are currently entering the second stage by defining the conceptual model according to the elicited ontological requirements.

Requirements specification. The requirements specification should involve the collaboration between domain experts, users, and the ontology engineers [6]. We organised a sequence of hands-on workshops to collect inputs from domain experts and expected users of the three use cases described in Section 2. As defined by the methodology [5], the first workshops focused on the *use case specification* and the discussion of the available documentation in the considered domain of knowledge (*data exchange identification*). Such documentation includes procedures in the form of text and tables in various formats (e.g., PDF, Excel), procedure-related manuals, images and videos, examples of procedure executions.

As a result, we defined the purpose and scope of the ontology, and we collected inputs to elicit more granular requirements. In particular, each use case defined a list of capabilities, representing the features that the PKMS solution should provide to the users. Such capabilities were defined considering a set of as-is and to-be scenarios and, besides providing technical requirements for developing the PKMS, were used as user stories for the identification of the ontological requirements. A series of workshops was dedicated to collaboratively deriving from each of these capabilities and, considering the documentation provided, a set of questions that the PKMS should be able to answer. Such questions have been then refined by us, in the role of ontology engineers, in the form of competency questions (CQs), that is, queries that the ontology to be developed should answer in order to retrieve relevant information from the data modelled with the ontology [7]. Each competency question has been then discussed with, and validated by, the domain experts.

Along with the CQs, as ontology engineers we also defined an initial proposal of *facts*, that is, natural language statements describing the domain to be represented in the ontology, possibly associated with a domain-specific terminology (e.g., attributes describing a specific term to be used in the ontology). Facts and CQs defined so far represent the initial proposal of ontological requirements for procedural knowledge representation. Such initial set of requirements includes 48 competency questions and 63 facts. Future iterations may lead to the definition of additional CQs and facts[6].
Let us make an example.

> **PKMS requirement.** "A procedure may have different steps and/or sub-procedures and may refer to other data sources of the company (e.g., documents/image/video) associated with it."
>
> **Competency Questions.** Two of the CQs that correspond to the requirement above are: "Which are the steps of a procedure?", and "From which data source a piece of information associated with a procedure has been collected?".
>
> **Facts.** Two facts that correspond to the CQs above are: "A step of a procedure can be either atomic, or decomposed into a subset of multiple steps", and "A step can refer to one or more resources (e.g., documents, media)".

Concepts and conceptual areas. The collected competency questions and facts identified by the domain experts and the ontology development team have been used in combination for extracting a set of relevant concepts and relations that build the procedural knowledge conceptual model, as an *informal* model. Such a model is a preparatory artefact for the ontology implementation activity [5] that develops the actual ontology using a formal language and reusing existing relevant ontologies.

While identifying such concepts, we clustered the requirements to obtain a conceptual representation of the domain into separate coherent subdomains based on conceptual areas. The criterion for defining

[5] https://lot.linkeddata.es/

[6] All materials will be made available on GitHub: https://github.com/perks-project/pk-ontology

such conceptual areas and their granularity varies depending on the ontology project, the specific ontology modelling choices, and the size of the domain.

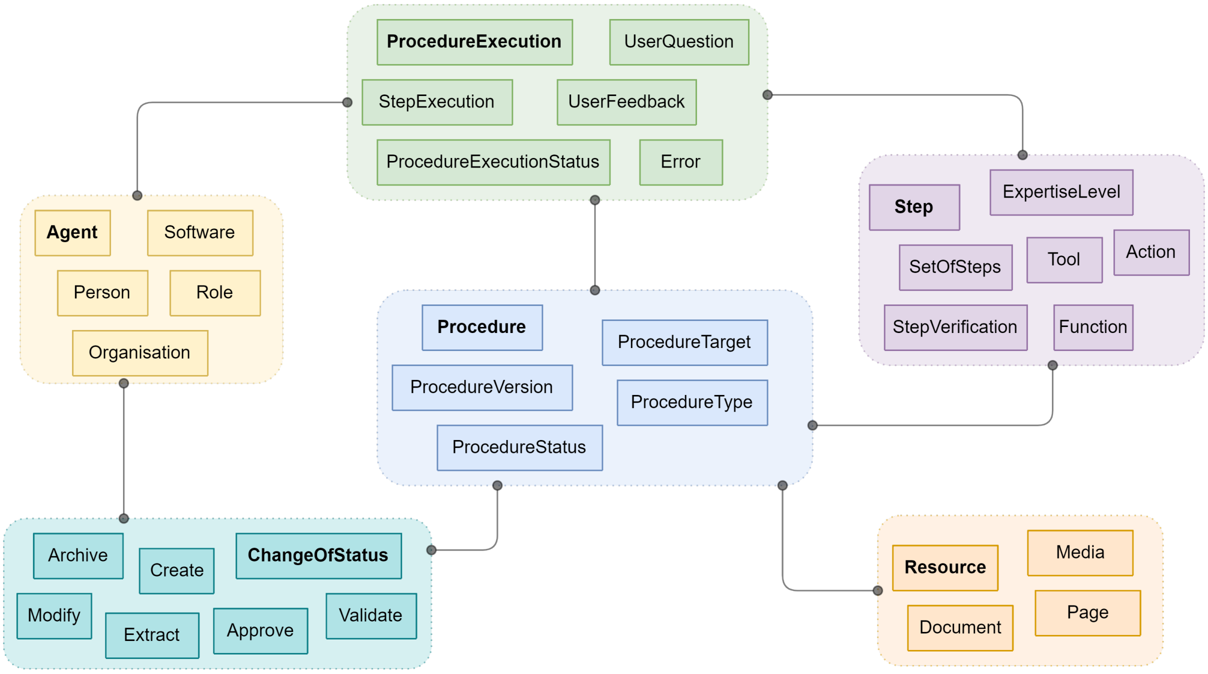

Figure 1: High-level overview of concepts and conceptual areas as a basis for the ontology of PERKS.

Figure 1 depicts the initial set of concepts that we identified as core to be included in the model, based on the requirements, and taking into account the technical solutions that will compose a PKMS. We came up with six conceptual areas that cluster our concepts, each having a representative concept that we can use for naming the subdomain: *Procedure*, *Step*, *Change of Status*, *Procedure Execution*, *Resource*, and *Agent*. Such conceptual areas are also put in relation by means of edges, which stand for relevant connections between concepts across areas.

Procedure. A `Procedure` is a sequence of actions to be executed in order to achieve a desired outcome. Such outcome can be expressed in terms of a `Procedure Type` (e.g., a maintenance activity), that is the type of procedure being executed on a `Target` (e.g., a production line for washing machines). A `Procedure` should be associated with at least one `Procedure Version`, that is a description of a specific `Set of Steps` for the execution of the `Procedure`. Based on our requirements, it appears clear that e.g. a procedure for maintenance of a specific production line for washing machines can be updated and revised, thus creating multiple versions. A `Procedure Version` is associated with a current `Status` (e.g., *draft*, *validated*, *approved*).

Change of Status. We want to keep track of the possible `Changes of Status` of a procedure, which result from some actions that an agent can perform on the procedure, in order to track provenance information that makes auditing operations possible. Examples of such operations are: `Create`, `Extract` (e.g., when a procedure is extracted from some textual document), `Modify`, `Validate`, `Approve`, `Archive` (when a procedure becomes obsolete and is replaced by a new version).

Step. A `Step` groups one or more `Actions` (from a human) or `Functions` (from an algorithm/software) to execute a portion of a `Procedure`, possibly with the use of some `Tools`, and the `Set of Steps` for executing the whole procedure corresponds to a specific `Procedure Version`. Moreover, a step can be decomposed as a `Set` of multiple `Steps` itself. From our use cases, it emerged the need of specifying within the procedure a `Step Verification`, that is the way in which the actual execution of a step can be verified. Finally, since different agents (e.g., junior *vs* senior technician) can perform the same procedure, we need the concept of `Expertise Level` to be associated with each (set of) steps(s), meaning that the step is targeted at an agent with that level of expertise, and allowing for different *formulations* and levels of detail of the same steps.

Procedure Execution. A certain Procedure can be executed one or multiple times, by one or more agents, at a certain time. This is represented by the concepts of `Procedure Execution` and `Step Execution`, thanks to which we can track the execution of the procedure as a whole, and as a composition of executions of the individual steps. As the Procedure Version, the execution can be associated with a `Status` too, e.g., *in progress, completed, paused.* It may happen that the agent executing the procedure wants to leave a `Feedback` about the procedure version or its execution, e.g., suggesting to add some details to the procedure based on its experience while executing it. Also, she may also have some questions or doubts that she had to solve while performing some steps. Finally, an `Error` can occur at some point, and the agent may search for a specific solution. It is important to keep track of such feedback/doubts/errors, so that the agent can be supported in solving them, and such pieces of information can be used to improve the procedures.

Agent. As it can be noticed, the `Agent` is a very important entity in the context of procedures and procedure executions. An agent can be a `Person`, an `Organisation`, a `Software`. Any agent, while interacting with a procedure, can play a certain `Role` (e.g., editor, supervisor, user).

Resource. A Procedure can refer to different `Resources`, like `Documents` (`Pages`) or `Media` that constitute relevant documentation; it can be extracted from a resource, as in the case of e.g. a PDF containing some text describing a procedure to execute, to which machine learning techniques can be applied to extract all the steps. A Procedure can be considered as a `Resource` itself, intended as any entity that can be included in a catalog.

Two additional relevant concepts, that can be considered orthogonal to the conceptual areas identified so far, are `Time` and `Sequence`. `Time` is here intended as both the date and time on which something occurred (e.g., the creation of a new version of a procedure, or its execution by a certain agent), and the duration of something (e.g., the amount of time that is expected to be needed for executing a step). `Sequence` stands for a set of related entities that follow each other in a particular order, like the sequence of steps to be executed one after the other, or the sequence of versions of a certain procedure after multiple updates by different agents.

Towards the ontology formalisation. As previously explained, the concepts and possible relations identified in this initial version of the procedural knowledge conceptual model will not be directly translated into an ontology model. Indeed, as recommended in [5], when entering the *ontology conceptualization* and *ontology encoding* phases of the methodology, we will reuse available ontological resources that solve the same, or similar, problems we need to address, in order to foster interoperability and facilitate knowledge reuse [8]. We already started the activity of scouting ontologies related to the procedural knowledge domain, and we are taking into account different sources, including papers that analyse the state of the art, such as [9], and general (e.g., LOV[7]) and domain-specific (e.g., the Industry Portal[8]) ontology repositories. Among others, we are considering existing ontologies like P-Plan [10] (which models plans and their steps) and PROV-O[9] (for provenance tracking) and existing well-known languages like BPMN[10] (which represent business processes); we will evaluate their ability to partially cover our requirements for the semantic description of both procedures and other resources associated with them, as well as procedures' life cycle and actual executions.

Such analysis of existing ontologies, and the resulting selection of ontologies to be reused, is an ongoing activity. The resulting ontology will be openly released and published following FAIR principles and web ontology publication best practices [11].

[7] https://lov.linkeddata.es/

[8] https://industryportal.enit.fr/

[9] https://www.w3.org/TR/prov-o/

[10] https://www.omg.org/spec/BPMN/2.0/

4. Conclusions

In this paper, we presented our activities aimed at supporting the holistic governance of industrial procedural knowledge in its life cycle, including its elicitation, management, access and exploitation, by leveraging AI and data technologies. Our work so far consisted in two main activities are: (i) the collaborative requirements collection from three heterogeneous real-world use cases, and, based on such requirements, (ii) the analysis of the domain to be modelled in the ontology in the form of a set of concepts, distributed into conceptual areas that represent sub-domains. These activities are preliminary to the development of an ontology for modelling procedural knowledge in Industry 5.0, which will be the basis of the modular architecture for PK management (PKMS) we are defining. In future work, we plan to continue the work on the ontology for its formal implementation, publication and evaluation, and its employment in demonstrators to support the three presented use cases. Moreover, we plan to engage with additional stakeholders to evaluate the transferability of the defined model and the overall PKMS solution to additional PK use cases. In this direction, we established a *user board* involving additional stakeholders not directly involved in the three industrial scenarios, but who may be facing similar challenges with PK management.

Acknowledgments

This work is partially supported by the PERKS project, co-funded by the European Commission (Grant id 101070186).

References

[1] Y. Zhou, J. Shah, S. Schockaert, Learning Household Task Knowledge from WikiHow Descriptions, in: L. Espinosa-Anke, T. Declerck, D. Gromann, J. Camacho-Collados, M. T. Pilehvar (Eds.), Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5), Association for Computational Linguistics, Macau, China, 2019, pp. 50–56. URL: https://aclanthology.org/W19-5808.

[2] L. Zhang, Reasoning about Procedures with Natural Language Processing: A Tutorial, 2022. doi:10.48550/arXiv.2205.07455, publication Title: arXiv e-prints ADS Bibcode: 2022arXiv220507455Z.

[3] P. Bellan, M. Dragoni, C. Ghidini, Extracting business process entities and relations from text using pre-trained language models and in-context learning, in: International Conference on Enterprise Design, Operations, and Computing, Springer, 2022, pp. 182–199.

[4] M. Hulea, et al., PERKS project deliverable D4.1: Reference architecture for a Procedural Knowledge Management System, 2024. URL: https://zenodo.org/communities/perks_project, (to appear).

[5] M. Poveda-Villalón, A. Fernández-Izquierdo, M. Fernández-López, R. García-Castro, LOT: an industrial oriented ontology engineering framework, Eng. Appl. Artif. Intell. 111 (2022) 104755. URL: https://doi.org/10.1016/j.engappai.2022.104755. doi:10.1016/J.ENGAPPAI.2022.104755.

[6] M. Scrocca, I. Baroni, I. Celino, Urban IoT ontologies for sharing and electric mobility, Semantic Web 14 (2023) 617–638. URL: https://content.iospress.com/articles/semantic-web/sw210445. doi:10.3233/SW-210445, publisher: IOS Press.

[7] M. Grüninger, M. S. Fox, The role of competency questions in enterprise engineering, in: Benchmarking—Theory and practice, Springer, 1995, pp. 22–31.

[8] V. A. Carriero, M. Daquino, A. Gangemi, A. G. Nuzzolese, S. Peroni, V. Presutti, F. Tomasi, The landscape of ontology reuse approaches, in: Applications and practices in ontology design, extraction, and reasoning, IOS Press, 2020, pp. 21–38.

[9] A. Harth, T. Käfer, A. Rula, J. Calbimonte, E. Kamburjan, M. Giese, Towards representing processes and reasoning with process descriptions on the web, TGDK 2 (2024) 1:1–1:32. URL: https://doi.org/10.4230/TGDK.2.1.1. doi:10.4230/TGDK.2.1.1.

[10] D. Garijo, Y. Gil, Augmenting PROV with plans in P-PLAN: scientific processes as linked data, in:

T. Kauppinen, L. C. Pouchard, C. Keßler (Eds.), Proceedings of the Second International Workshop on Linked Science 2012 - Tackling Big Data, Boston, MA, USA, November 12, 2012, volume 951 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2012.

[11] D. Berrueta, J. Phipps, A. Miles, T. Baker, R. Swick, Best practice recipes for publishing RDF vocabularies, Technical Report, W3C, 2008. URL: https://www.w3.org/TR/swbp-vocab-pub, [Accessed September 2024].