Providing domain knowledge for process mining with ReWOO-based agents

Max W. Vogt[1], Peter van der Putten[1], and Hajo A. Reijers[2]

[1] LIACS, Leiden University, the Netherlands
[2] Department of Information and Computing Sciences, Utrecht University, the Netherlands

Abstract. Process mining practitioners often face the challenge of interpreting complex process data and driving process improvements with limited expertise in process optimization, tools, and the application domain of the process. This study explores the integration of LLM-based agentic frameworks in process mining to bridge this gap and democratize access to process optimization. We developed a Proof-of-Concept that leverages a Reasoning WithOut Observation (ReWOO)-based agent to perform process discovery, problem identification, generate ecosystem domain knowledge, and propose potential process improvements. Our experiments on a range of business processes suggest that LLM-based agent systems can insert meaningful domain knowledge into process mining tool interactions.

Keywords: Process mining · LLM · ReWOO · Domain knowledge

1 Introduction

Process mining offers several benefits for organizations, such as providing fact-based insights into processes, aligning with process models, and improving processes. These benefits can improve productivity and business outcomes, and reduce risks. However, organizations often encounter obstacles when trying to derive value from process mining. One issue Zerbato et al. found in an overview of best practices is the lack of domain knowledge, which is often encountered by practitioners of process mining [17]. Similarly, Andrews et al. found that process improvements are largely proposed by domain experts and that the process mining tools themselves do not provide direction for these improvements [1].

We propose to explore the potential of using LLMs to provide domain knowledge in process mining tools. We validated the lack of domain knowledge in process mining and the potential of using LLMs to fill this gap through in-depth interviews with a small set of process mining practitioners.

Several studies have been done about using LLMs in combination with process mining, such as connecting the open-source process mining tool PM4PY to an LLM [2]. Not only academics are looking into using LLMs in process mining, but we also see interest from commercial companies. For instance, process mining

vendors Celonis[3] and Pegasystems[4] are currently developing LLM capabilities in their process mining solutions. However, these efforts are mainly focusing on using LLMs for descriptive tasks in process mining tools, e.g. 'How long does my process take?' or answering questions regarding process mining expertise, e.g. 'What is conformance checking?'.

We propose a different approach. We use LLMs-based agents to fill the observed gap in domain knowledge in process mining tools, the type of knowledge that would traditionally come from a human domain expert, e.g. knowledge about the sector of the organization. Hence our research question is, 'How can LLMs provide domain knowledge to help users of process mining tools understand and improve processes?'. To study this, we follow the principles of a research by design study. [11]. We developed a Proof-of-Concept (PoC) and evaluated this with a limited set of experiments. The results indicate that LLM-based systems can generate domain knowledge and give directions for future work.

The remainder of this paper is as follows: Section 2 discusses background and Section 3 related work. Next we present our conceptual vision (Section 4), the PoC (Section 5) and evaluation (Section (6), followed by a discussion of limitations and future work (Section 7) and a conclusion (Section 8).

2 Background

In this section, we discuss LLM-based agents, and the ReWOO agentic framework as a special case.

2.1 LLM-based agents

AI agents are systems that can make decisions and take actions, based on their perception of their environment [16]. The development of LLMs led to an increasing interest in LLM-based agents, as these models allow for the creation of more powerful agents. LLM-based agents can interact via natural language with a human user, making the agent easier to use and more explainable [13], and can leverage generative capabilities to create plans, understand tool capabilities, build up context, and interpret feedback.

2.2 Reasoning WithOut Observation

Reasoning WithOut Observation (ReWOO) is an LLM-based agent framework proposed by Xu et al. as an improved version of Reasoning and Acting (ReAct) [15]. ReAct proposes that the agent has reasoning steps in between the execution of tools and then observes the generated output [16]. While this approach allows

[3] https://accelerationeconomy.com/ai/process-mining-meets-generative-ai-celonis-rides-industry-wave-to-democratize-core-tech/

[4] https://www.pega.com/about/news/press-releases/pega-launches-pega-process-mining-generative-ai-ready-apis-enable,https://docs.pega.com/bundle/process-mining/page/process-mining/content/whats-new.html

an agent to solve more complex tasks there are some drawbacks. For instance, ReAct calls the LLM at each step of the process, creating more computational complexity and increasing token usage.

ReWOO decouples the reasoning and acting parts of the agent. A ReWOO agent consists of three main instances: a (1) planner, (2) worker, and (3) solver. The planner creates a plan for the execution of the task provided by the user, the worker iteratively executes the tools of the agent based on the generated plan, and the solver generates the final output and returns that to the user. According to experiments by Xu et al. ReWOO achieved a 5x token efficiency and 4% accuracy improvement compared to ReAct.

3 Related work

Both business and academia have been exploring the use of LLMs in process mining, such as using LLMs to interpret process models via textual abstractions of these models [2, 3]. This approach allows the model to interpret the process model and answer user queries about it, but it does not utilize more advanced prompt engineering techniques and uses a 'zero-shot' prompting approach. Busch et al. provided an overview of the potentials and challenges for using prompt engineering techniques in BPM [4], and Jessen et al. developed an approach for using LLMs in process mining, aiming to make process mining tools more accessible [7]. Their approach focuses on using LLMs to create a conversational agent that can answer questions about the process model or event log.

Eichele et al. studied the potential of adding domain knowledge to event logs, using Web Ontology Language (OWL) ontologies [5]. With this approach, they can map the domain knowledge to specific cases and activities. This angle differs from our approach as it focuses on the manner of supplementing domain knowledge and not on the automatic generation of domain knowledge. Therefore, these two different angles can potentially benefit from each other.

Current studies are focused on using LLMs in process mining tools for generating process models, providing process mining and process domain knowledge. Our approach proposes to use LLMs to provide another type of domain knowledge into process mining tools: delivering specific domain knowledge, for instance, based on characteristics of the sector or the specific organization, that is relevant to the generated process model. The provided domain knowledge allows a user to interpret the process model or make process improvements, making process mining tools more user-friendly, time-efficient, and accessible.

4 Conceptual vision

In this section, we first present the different types of domain knowledge that we distinguish. Then we present our conceptual framework that shows where and how domain knowledge, generated by an LLM, can be incorporated into a process mining tool to solve the observed scarcity of domain knowledge.

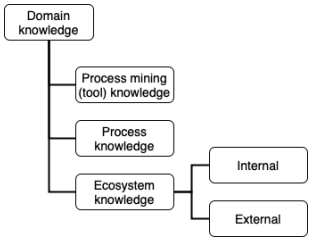

Fig. 1. Types of domain knowledge

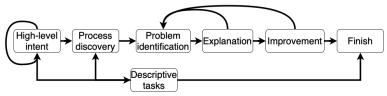

Fig. 2. Proposed conceptual framework

4.1 Domain knowledge

Domain knowledge in process mining is knowledge about the context of the process and knowledge that is necessary to analyze and fully understand the generated processes [5]. We make the distinction between three types of domain knowledge (Figure 1). Firstly, process mining knowledge: Knowledge about the domain of process mining itself, which can be an inaccessible field for non-experts [8]. So this can include knowledge about the use of a specific process mining tool.

Secondly, process optimization knowledge: Knowledge about process optimization, regardless of the process mining tool used, and mostly independent of the specific problem domain or industry. E.g., knowledge of Lean Six Sigma, a set of techniques for process improvements and waste elimination [18].

Thirdly, ecosystem knowledge: Knowledge about the domain and business of the organization and the ecosystem surrounding the process (e.g. regulation). This knowledge is important when interpreting the process model and when trying to make changes to the process. The lack of this knowledge has been identified in practice as an issue in current implementations of process mining [9]. Within this type of domain knowledge, we make the distinction between external and internal. Internal ecosystem domain knowledge consists of knowledge only available within an organization, e.g. internal documentation of an organization. External ecosystem domain knowledge is publicly available knowledge, e.g. regulatory documents or sector reports published online.

4.2 Framework

The framework (Figure 2) consists of seven main components. The first one is the 'high-level intent', the user specifies what the goal of the analysis is and potentially what tools should be used. This can be done by using a single prompt or by having a conversation with the system. In this conversation, the user can express the desired output and usage of tools, ask clarifying questions, and the system can inform the user about the available options. Furthermore, the system can ask questions to the user to allow for the most complete picture of the high-level intent, e.g. asking the name of the organization or the type of process.

In the 'process discovery' component, the system will generate the process model from the event log that was provided by the user. If the user specifies a

preference for a specific approach, that approach will be used. Otherwise, the system will pick an approach, based on process mining domain knowledge.

The third component of the framework is 'problem identification', the system will analyze the generated process from the previous step and identify the relevant information. What this relevant information is and how it is identified depends on the preferences of the user from the high-level intent. If the user does not specify a preference, the system will determine what type of analysis should be used, based on ecosystem domain knowledge. For instance, for spotting process inefficiencies, an analysis based on the frequency and duration of the process and its activities could be the best option.

In the fourth component, the 'explanation', the system generates relevant ecosystem domain knowledge for the user based on the results of the problem identification phase. For instance, if the problem identification phase has identified inefficient process components, the explanation phase can provide explanations about why these inefficiencies might be happening in the process. Based on the explanations that the system provides, the user can refine the problem identification, e.g., by applying a filter.

The fifth component is 'improvements', an extension of the explanation phase where the system will generate potential solutions addressing the identified problems in the process (e.g. improvements for removing bottlenecks). The user can refine the problem identification during this phase, e.g., by letting the system update the process with the proposed process improvements and execute problem identification on the updated process. If the user does not want to refine the problem identification and is content with the output from the system, the final result will be generated and returned to the user (sixth component, 'Finish').

Besides the mentioned components, there is another one that can be used, the 'descriptive tasks' component. This represents the option for the user to ask descriptive questions about the event log (before process discovery) or generated process model (after process discovery). For instance, 'How many activities are between A and B in the process model?'. These are the types of questions that can be answered with process mining domain knowledge and process domain knowledge. The capacity to answer these types of questions with LLMs is already being studied and developed by others (e.g., [2, 7]).

With this framework, we allow the execution process mining techniques and incorporate the different types of domain knowledge. During the execution of the system, the user can still interact with the system. For instance, to change the type of analysis or ask clarifying questions. This design should help to fill the mentioned shortage of domain knowledge in current process mining tools.

5 Proof of concept

Based on the conceptual vision and framework, we developed a PoC to explore the value of a system that provides domain knowledge in a process mining tool.[5]

[5] For links to architecture, code, input event logs, experimental set up and outputs of experiments see https://github.com/maxvogt12/ReWOO_agent_for_PM

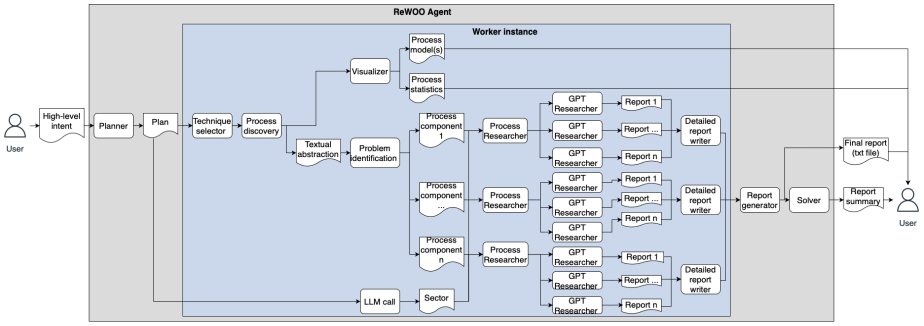

Fig. 3. Architecture flow of the PoC

The framework is a vision of a system that can execute all process mining techniques integrating all types of domain knowledge (Figure 1), but for our PoC we focused on a process mining system integrating ecosystem domain knowledge, as the generation of this type of knowledge has not been studied yet. To keep the approach generic, we zoomed in on external ecosystem domain knowledge (through GPT Researcher, see below). This approach could easily be extended by adding documentation internal to an organization. Our PoC contains all components of the framework (Figure 2) except 'descriptive tasks' and all the code is ours (except the mentioned libraries). Requirements for the PoC were gathered through a small set of in-depth interviews with process mining practitioners.

We use a ReWOO-based agent to let the system execute external functions (e.g. process mining techniques) and solve more complex tasks. We use ReWOO to lower the computational complexity and costs of running the agent, while maintaining the quality of performance (compared to ReAct). When the agent receives the high-level intent from the user, it will generate a plan. The generated plan consists of Plan and #E steps. Plan steps contain the goal of the step and the #E step contains the tool with the input the agent shall use to reach this goal. The outcome is captured in the #E variable and can be used in later steps. E.g. the first step of such a plan: 'Plan: Create a process model using the DFG approach and identify the top 3 IT and cyber risks. #E1 = ProcessDiscovery_DFG[file_path]'.

With the available tools, the agent can fulfill the steps in our framework (Figure 2). From the user (in the 'high-level intent'), the system requires the file path to the event log, the name of the organization, the type of the process, and the type of analysis that should be conducted. The user can also express a preference for the approach that should be used for problem identification (DFG, temporal profile, or variants), otherwise, the system will pick one (Technique Selector tool). Our agent has the following tools at its disposal.

The **Process Discovery** tool generates a process model from an event log ('process discovery' and 'problem identification' in Figure 2). It uses the textual abstraction functionalities of PM4PY [3], visualizes the process model in multiple

formats (DFG, Petri net, BPMN, heuristic net, and decision tree) and provides statistics about the event log, like the distribution of events over time.

The **Technique Selector** selects the appropriate basis for the problem identification technique (DFG, temporal profile, or variants) for the analysis that the user has requested, e.g. focusing on process inefficiencies. The user can request a specific technique that should be used by the agent. If the user does not specify a preference, the tool uses internal ecosystem domain knowledge in the form of a document ('Prototype/Tools/Process_Mining/PM_approach_mapping.txt' on GitHub) that the tool analyses to base its decision on, simulating the use of internal documents of an organization. If the document does not contain any useful information to base the decision on and the user does not provide a preference the agent will pick the default technique (DFG).

The **Process Researcher** tool generates research reports about the identified process components, supplying explanations and potential improvements for the identified process issues ('explanations' and 'improvements', Figure 2). For each provided process component one report is generated, based on external ecosystem domain knowledge from the internet. Reports are generated by the GPT Researcher module, an open-source project of an LLM-based research assistant [6], the information within the reports is referenced to reduce hallucination. The reports can have two levels of detail: standard (one GPT Researcher run, inspired by the Plan-and-Solve approach [14]) or detailed (multiple GPT Researcher runs, inspired by the STORM approach [12]).

The **LLM** allows the agent to ask simple questions to an LLM, e.g. to find the sector of an organization.

The **Human-in-the-loop** allows the agent to interact with the human user for input if it requires that, the response of the user is then used in the execution of the agent.

The **Report Generator** generates the final report, which aligns with the 'explanations' and 'improvements' steps of the framework (Figure 2) as this report includes potential explanations and improvements. The final report is returned as a separate text file because these files are too big to be returned to the user as the output of the agent.

Figure 3 shows an overview of the sequential steps of an execution of the agent using the mentioned tools (see the GitHub for a more detailed description of these steps), although this is not a static order of steps as the agent constructs a plan of action each time it is run. If the constructed plan is insufficient for solving the task, the agent can go back to the planner instance and change it. This type of looping within the agent is controlled through a graph structure [19].

The PoC is based on the LangChain[6] framework, because this framework is open-source, regularly updated, and offers a versatile set of functionalities. We use OpenAI's, GPT-3.5 for simple tasks that do not require a lot of input and output tokens and GPT-4 for the more complex tasks that require a lot of input and output tokens [10]. We chose these models, as they are well-integrated into the LangChain framework and GPT Researcher module.

[6] https://www.langchain.com

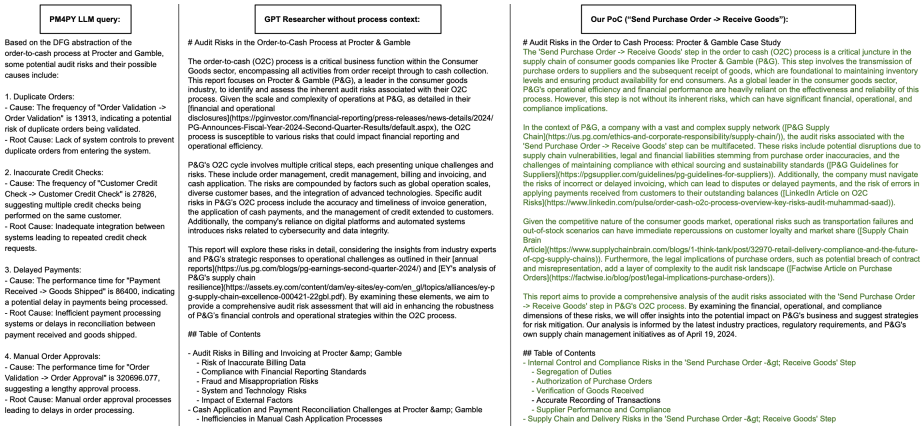

Fig. 4. Sample from the generated reports from the experiments

6 Evaluation

In this section, we discuss the methods we used to evaluate the PoC. We conducted two experiments: an ablation experiment and a qualitative analysis of the output of our PoC across different executions (see Github for full details).

6.1 Ablation experiment

We tested three different ablations to compare their outputs. The outputs that were generated are too large to fully display in this paper, therefore, we present the first part of the outputs. The three ablations that we tested are: (1) the PM4PY LLM query functionality [2], (2) GPT Researcher [6], and (3) our PoC. We selected these three systems because PM4PY represents the process mining capabilities and GPT Researcher the domain knowledge generation capabilities of our PoC. We compared these two to our PoC to see if the construction of our PoC offers additional value compared to these separate components of our PoC.

We used the same prompt for all three systems, asking what the audit risks are in an order-to-cash (O2C) process at Procter and Gamble (P&G), and what could be causing them. The dataset that we used can be found on our GitHub page ('/Event_Logs/O2C.csv'). The data is about an O2C process but not specifically at P&G, we added this for testing purposes. The beginning of the output is shown in Figure 4 and our analysis of the output is shown in Table 1.

The PM4PY LLM query system was able to execute process discovery and problem identification but could not generate in-depth domain knowledge. GPT Researcher was able to generate more in-depth domain knowledge about O2C processes. However, because it cannot execute process discovery and problem identification, the domain knowledge is superficial in the sense that it is not focused on specific activities or parts of the O2C process. Our PoC combines the strengths of these two systems, leading to in-depth domain knowledge that

Table 1. Analysis of the output of the ablation experiment

Ablation	Analysis of output (see Figure 4 for output)
PM4PY LLM Query	The textual abstraction (DFG) allows the LLM to analyze the process model and reason which activities in the process could form an audit risk based on the DFG (process domain knowledge). However, the explanations about the causes of these audit risks (ecosystem domain knowledge) are superficial compared to the other systems.
GPT Researcher	Generates more in-depth knowledge for the provided question. However, it is unaware of the components of the system and that makes the answers more superficial, the information is just about an O2C process but not about specific components within this process. GPT Researcher provides references for the information that it generates.
PoC	Executes process discovery and problem identification and can then provide relevant process components to the GPT Researcher. This allows for the generation of in-depth knowledge that is focused on the relevant parts of the process, in this case, the 'Send Purchase Order ->Receive Goods' step. Our PoC provides references for the information that it generates.

is focused on a specific part of the O2C process that was identified using process discovery and problem identification. The beginning of this report is shown on the right in Figure 4; we marked all information green that was specific for the process component that was passed along, compared to the 'generic' report of the GPT Researcher ablation (middle part of Figure 4).

In the output of our PoC, we see more detailed information, e.g. focusing on a specific risk for receiving goods: "Given P&G's global operations, purchase orders often involve transactions in foreign currencies. Fluctuations in exchange rates can significantly impact the cost of goods received, leading to financial statement volatility.".

The system can combine information about the organization with current trends to identify risks for P&G. The report mentioned that P&G is active in over 70 countries and therefore faces geopolitical risks like the war in the Middle East and the Houthi attacks on cargo ships. The used source is an example of RAG, the article[7] is from February 2024 and the pre-trained knowledge of the GPT-4 model (currently) is cut off in December 2023 [10]. The reports also include potential improvements for the observed audit risks in the process, e.g. that P&G can use the Third-Party Risk Management (TPRM) process and internal control frameworks that it has in place to mitigate some of the mentioned risks (including a reference to the website mentioning this framework).

Another ablation experiment was carried out asking what the risks are in context of an automotive IT incidents handling process, with similar results, for instance pointing out that an observed delay in assessing a recorded incident leads to increased cyber security risks (see Github for full results).

6.2 Qualitative analysis

With the qualitative analysis, we evaluate if the PoC performs differently across various problem identification types or use cases compared to others. This is important for determining whether or not the PoC offers additional value in all areas or specific ones. Furthermore, we use different process types (event logs can

[7] https://sphera.com/spark/top-10-operational-risks-for-2024/

Table 2. Compared results of our PoC

Problem identification	Sector/ use case	Process type	Technique	Average
Environmental risk	Retail	Purchase-to-Pay (P2P)	DFG	92.58%
Audit risk	Consumer goods	Order-to-Cash (O2C)	DFG	90.07%
IT & cyber risk	Automotive	IT incident handling	Temporal profile	82.10%
Process inefficiencies	Manufacturing	Accounts Payable (AP)	Variants	74.37%
Operational risk	Technology	Travel expenses	Variants	68.64%
Regulatory risk	Financial services	Loan application	Temporal profile	66.96%

be found on GitHub at 'Experiments/Prototype_exp') and different techniques and execute multiple runs to see if this influences the results.

We let our PoC generate a report and then we analyzed how much of the content of the report contains information that is specific to the process component that the system analyzed. We used this approach since LLMs and LLM-based systems tend to generate generic knowledge and therefore a challenge is to let such a system generate specific information. Generic information could be removing manual processing (can be applied to many processes and activities). Process (component) specific information could be the impact of return policies on the carbon footprint of an organization ('Return Goods' step in a P2P process). For this experiment, process component-specific information is information where the report names the process component or argues about it. We used this same approach in Figure 4, the green marked text on the right indicates process (component) specific. As an extension, it would be useful to study the portion of the report that contains useful domain knowledge (see Section 7).

Table 2 shows the average results of the qualitative analysis for all the reports we generated (multiple reports per row, see GitHub). We observe that the average scores for Audit risk and Environmental risk are higher than those for Process inefficiencies and Regulatory risk. This could be because there are more online sources available about these problem identification types, the organization, or the process type. Another interesting thing is that the higher-scoring ones both used the DFG technique, while the other two did not. We see that all averages are above 65%, so 65% of the report contained specific information. For the full reports, see the 'Experiments/Prototype_exp' folder on GitHub.

Overall, the ablation experiment showed that our PoC offers additional value over some of its components, PM4PY LLM query, and GPT Researcher. The qualitative analysis indicated that within all reports across different problem identification types, over 65% of the content represented information (external ecosystem domain knowledge) specific to the process component(s).

7 Discussion, limitations and future work

Although our research presents a novel way to generate domain knowledge within process mining tools, it has its limitations. First, as the goal was to design and deliver initial proof for the concept, we haven't run a full end-user test with

domain experts. Iterative development and validation with an end-user group would be an interesting next step.

Our PoC works best for organizations with a lot of online resources, although the industrial sector is used if the organization itself is not known. If this type of system were to be used in a real-life it could rely on documentation of the organization (internal ecosystem domain knowledge) to fill this gap. We demonstrated this by using a document as the information source for the Technique Selector tool.

The PoC bases the generated domain knowledge on the outcomes of the process discovery and problem identification phases but does not allow refinement of the problem identification phase based on the generated domain knowledge. We did include this option in our conceptual vision (Figure 2) but it would be interesting to develop this functionality, where the user could demand a new analysis (e.g. filtering) based on the generated domain knowledge.

For the experiments, we assessed three ablations of the system and conducted a qualitative analysis on multiple reports. To get a more complete picture of the added value of our approach, it could prove useful to conduct user tests with people who do and who do not have a lot of process mining experience.

The output of the PoC is rather large (often 10+ pages) as we tried to generate as much relevant domain knowledge as possible to study the potential of LLMs for this task. However, for a business user, it might prove useful to investigate a filter for this large output and to just return the most useful output for a specific user. This can be realized by changing the prompts of the 'Solver' instance (Figure 3), to let it write a summary.

8 Conclusion

In this paper, we have explored if and how LLMs can improve the current state of process mining by generating domain knowledge for process mining tools. We have presented a conceptual vision that shows the design of such a system and its components. Our PoC is a tangible system that demonstrates that our vision can already be developed and offer additional value compared to current LLM implementations in process mining. Through our experiments, we observed that our PoC provided more in-depth external ecosystem domain knowledge about specific process components compared to the other systems in the experiment. We believe that if the mentioned challenges are overcome, our proposition can offer even more value to process mining practitioners.

References

1. Andrews, R., Wynn, M.T., Vallmuur, K., ter Hofstede, A.H.M., Bosley, E.: A comparative process mining analysis of road trauma patient pathways. International Journal of Environmental Research and Public Health **17**(10) (2020)
2. Berti, A., Qafari, M.S.: Leveraging large language models (LLMs) for process mining (technical report) (2023), https://arxiv.org/abs/2307.12701

3. Berti, A., Schuster, D., van der Aalst, W.M.P.: Abstractions, scenarios, and prompt definitions for process mining with llms: A case study. In: De Weerdt, J., Pufahl, L. (eds.) Business Process Management Workshops. pp. 427–439. Cham (2024)
4. Busch, K., Rochlitzer, A., Sola, D., Leopold, H.: Just tell me: Prompt engineering in business process management. In: International Conference on Business Process Modeling, Development and Support. pp. 3–11. Springer (2023)
5. Eichele, S., Hinkelmann, K., Spahic-Bogdanovic, M.: Ontology-driven enhancement of process mining with domain knowledge. In: AAAI 2023 Spring Symposium on Challenges Requiring the Combination of Machine Learning and Knowledge Engineering (2023)
6. Elovic, A.: GPT Researcher. https://github.com/assafelovic/gpt-researcher?ref=blog.langchain.dev (2024), accessed on 26-04-2024
7. Jessen, U., Sroka, M., Fahland, D.: Chit-chat or deep talk: Prompt engineering for process mining (2023), https://arxiv.org/abs/2307.09909
8. Kampik, T., Warmuth, C., Rebmann, A., Agam, R., Egger, L.N.P., Gerber, A., Hoffart, J., Kolk, J., Herzig, P., Decker, G., van der Aa, H., Polyvyanyy, A., Rinderle-Ma, S., Weber, I., Weidlich, M.: Large process models: Business process management in the age of generative AI (2023), https://arxiv.org/abs/2309.00900
9. Martin, N., Fischer, D.A., Kerpedzhiev, G.D., Goel, K., Leemans, S.J., Röglinger, M., van der Aalst, W.M.P., Dumas, M., La Rosa, M., Wynn, M.T.: Opportunities and challenges for process mining in organizations: results of a Delphi study. Business & Information Systems Engineering **63**, 511–527 (2021)
10. OpenAI: Models. https://platform.openai.com/docs/models/overview (2024), accessed on 28-04-2024
11. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. Journal of Management Information Systems **24**(3), 45–77 (2007)
12. Shao, Y., Jiang, Y., Kanell, T., Xu, P., Khattab, O., Lam, M.: Assisting in writing Wikipedia-like articles from scratch with large language models. In: Duh, K., Gomez, H., Bethard, S. (eds.) Proc. NAACL (Vol 1: Long Papers). pp. 6252–6278. Mexico City, Mexico (Jun 2024)
13. Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., et al.: A survey on large language model based autonomous agents. Frontiers of Computer Science **18**(6), 186345 (2024)
14. Wang, L., Xu, W., Lan, Y., Hu, Z., Lan, Y., Lee, R.K.W., Lim, E.P.: Plan-and-Solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In: Annual Meeting of the Association for Computational Linguistics (2023)
15. Xu, B., Peng, Z., Lei, B., Mukherjee, S., Liu, Y., Xu, D.: Rewoo: Decoupling reasoning from observations for efficient augmented language models (2023), https://arxiv.org/abs/2305.18323
16. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., Cao, Y.: ReAct: Synergizing reasoning and acting in language models. In: International Conference on Learning Representations (ICLR) (2023)
17. Zerbato, F., Soffer, P., Weber, B.: Process mining practices: evidence from interviews. In: International Conference on Business Process Management. pp. 268–285. Springer (2022)
18. Zhang, Q., Irfan, M., Khattak, M.A.O., Zhu, X., Hassan, M.: Lean Six Sigma: a literature review. Interdisciplinary Journal of Contemporary research in business **3**(10), 599–605 (2012)
19. Zhuge, M., Wang, W., Kirsch, L., Faccio, F., Khizbullin, D., Schmidhuber, J.: Language agents as optimizable graphs. arXiv preprint arXiv:2402.16823 (2024)