Process Copilot: Speeding up Process Digitization in the German Public Sector

Birte Caesar[0000−0001−7772−7337], Solveig Peter[0009−0009−1391−3988], Celsa Diaz-Tejada[0009−0009−4788−0096], Matthias Werner[0000−0002−6142−1525], Oskar Krafft[0009−0006−6923−2290], and Michél Mahler[0009−0000−3847−0591]

PD-Berater der öffentlichen Hand GmbH, Friedrichstr. 149, 10117 Berlin, Germany
{firstname.lastname}@pd-g.de
https://www.pd-g.de/

Abstract. Public institutions seldom document the informal, undocumented processes that shape everyday work. As institutions digitize and modernize, such tacit knowledge is easily lost, particularly amid demographic change and limited process modeling expertise. To address this problem, we present Process Copilot, a generative AI system that converts natural language process descriptions into BPMN 2.0 models using large language models (LLMs). To improve the quality and reliability of generated process models, we incorporate domain-specific ontologies into the prompting process. We evaluate two prompting strategies, with and without ontologies, across ten real-world processes, using syntactic and structural similarity metrics alongside stability analysis based on output variance. Results indicate that ontology integration enhances semantic and structural performance, though at the cost of stability.

Keywords: Process Modeling · BPMN2.0 · Generative AI · Knowledge Graphs · Domain Ontology.

1 Introduction

The ongoing modernization of public administration compels institutions to digitize and streamline their internal processes [18]. Yet process knowledge remains fragmented across individuals, hindering transparency and documentation [22]. This problem is further exacerbated by demographic shifts and the loss of experienced staff [4]. While formal process modeling automation using standards like BPMN 2.0 [1] offers a potential solution, many public institutions lack the modeling expertise to standardize and implement such methods effectively.

Prior to LLMs, automation of knowledge graph generation relied on natural language processing (NLP) techniques like embedding-based and pre-trained language models (e.g., PG-BERT), which struggled with domain ambiguity and required large, annotated datasets, limiting scalability [12][23]. **LLM-based knowledge graph construction** offers a promising alternative, enabling the

[1] https://www.omg.org/bpmn/

joint extraction of entities and relations in a single step through methods such as in-context learning and fine-tuning [6], though fine-tuning is often impractical due to resource constraints [1]. In-context learning, while more accessible, faces challenges such as hallucinations, lack of implicit knowledge, and poor reproducibility [13]. Recent advances incorporate domain-specific ontologies as semantic guardrails to improve output quality, consistency, and reduce hallucinations in LLM-generated knowledge graphs [17][5]. **LLM-based process modeling** is closely linked to LLM-driven knowledge graph construction, as both require domain-specific knowledge and produce structured outputs like graphs or diagrammatic languages. To improve quality and reproducibility, Kourani et al. [14] proposed a prompting strategy combining role definition as a BPMN 2.0 expert and Chain-of-Thought prompting to extract intermediate POWL graphs, later converted to BPMN code outside the LLM. Neuberger et al. [19] experimented with embedding ontology-like information in prompts to structurally guide outputs, while Köpke et al. [16] introduced a meta-model intermediate representation similar to ontologies to reduce prompt complexity during iterative refinements. However, to the best of our knowledge, no process modeling approach to date integrates formal ontologies with LLMs.

Ontology-based Process Modeling aims to reduce errors in manual process design by enabling the creation of semantically precise, BPMN-compliant models using formal ontologies. While prior approaches either focus on modeling BPMN syntax or embedding domain-specific knowledge, they typically rely on expert users and do not leverage LLMs for automatic generation from unstructured text. To address this limitation, we extend **Process Copilot** – a generative AI-based system that transforms natural language process descriptions into BPMN 2.0 models – by integrating domain-specific ontology information into the prompting pipeline. This enables non-experts to produce formal models via a guided interface and LLM-powered transformation process. Building on prior work in LLM-guided modeling [9][10], we evaluate whether ontology integration improves the quality, consistency, and stability of the generated models. Our evaluation compares prompting strategies – with and without ontology integration – across ten real-world processes using structural and syntactic similarity metrics, as well as output variance.

2 Process Copilot

Process Copilot is an application to transform natural language process descriptions into BPMN 2.0-compliant process models using LLMs. The generated process models are presented in an interactive interface, allowing users to review, edit, and refine the diagrams as needed. Additionally, it includes functionality to export the resulting models in a format compatible with Camunda [2], enabling seamless integration into existing workflow management systems.

[2] https://camunda.com/platform/modeler/

LLMs for Process Modeling. As our work is conducted for the German public administration, we prioritize open-source LLMs that can be deployed on premise and perform reliably in German. We therefore tested a range of recent open-source models *Qwen 2.5 72B, LLaMA 3.3 70B, DeepSeek-R1 32B/8B, Qwen 2.5 32B/7B, Qwen 2.5-Coder 32B/7B, Mistral 3.1 24B, and Pharia-1 7B* for generating process diagrams in Mermaid syntax. Only the Qwen and Mistral LLMs consistently produced grammatically correct German Mermaid graphs without graph syntax errors (results are not part of this paper). While these analyses are ongoing, for the purpose of testing our central hypothesis, whether ontologies improve model outputs, we used OpenAIs *GPT-4o*, chosen for its stability, reproducibility, and strong performance under controlled hyperparameters ($max\_tokens = 1500$, $temperature = 0.1$, $top\_k = 50$, $top\_p = 0.9$). We excluded OpenAIs *o3* model because of lower reproducibility and a higher hallucination rate under comparable settings (results not published).

Prompt Engineering. We developed a hybrid approach that integrates established prompting techniques with domain-specific knowledge and empirical observations from previous work on administrative processes. Our strategy includes the following prompting methods: **(1) Role prompting**, assigning distinct responsibilities to the LLM across different phases to simulate expert behavior and improve task-specific reliability. **(2) Least-to-most prompting**, where simpler process models are generated and later integrated into more complex structures to support gradual model construction. **(3) Input enhancement**, where the initial prompt focuses on improving the quality of the user-provided natural language description to avoid inconsistencies during information extraction. **(4) Use of JSON as an intermediate representation**, serving as a bridge between text and BPMN 2.0. This choice leverages the overrepresentation of JSON in LLM training data (especially via web graph data such as JSON-LD) and facilitates conversion using open-source BPMN tools. **(5) Knowledge injection** via structured guidelines, including Chain-of-Thought prompting to guide the translation from text to JSON-based graph representations. **(6) Ontology-guided modeling**, using a custom RDF ontology to define the elements and evaluation criteria for process models.

The prompting methods outlined above are incorporated into the experimental prompts used to generate the process models (see Fig. 1), whose results are presented in the following section. The design of the methodology has been refined through multiple internal experiments using example cases tailored to our real-world public administration client needs. Nonetheless, we have identified potential areas for improvement, such as adding a dedicated BPMN 2.0 syntax correction role or a preprocessing step to handle inputs containing multiple processes. These points are discussed further in Section 4.

Ontology Usage. In this work, we investigate whether using an ontology as context in in-context learning improves model performance and stability compared to unstructured input. We adopt the same chain-of-thought prompting

```
1  role_definition: |
2  You are an expert in BPMN 2.0 and system architecture. Your task is to design a business
3  process model based on a textual process description.
4
5  instruction:
6  Task 1. Creation of a HIGH-LEVEL model by connecting suitable BPMN 2.0 elements.
7  Task 2. Creation of a DETAILED model by extending and configuring the elements.
8
9  requirements:
10 - The model MUST be created in JSON format that complies with the **JSON SCHEMA** specified
11   below.
12 - The process MUST begin with a `Start` event and end with one or more `End` events.
13 - ...
14
15 context:
16     component_library: ${Component Libary}
17     json_schema: ${JSON Schema}
18
19 input_data: ${Corrected Process Description}
```

Fig. 1: Excerpt of the used prompt

approach described previously but replace the component library with an ontology in RDF/Turtle syntax, which previous work has shown to be well-handled by LLMs. Instead of reusing an existing BPMN 2.0 ontology ([21]), we constructed a simplified version tailored to our application, omitting unnecessary details. As described above, we use a JSON-schema-based intermediate format, which later is transformed into BPMN XML format. The JSON schema's limited expressiveness restricts useful ontology modeling options, so we limited the ontology to classes, subclasses, and rdfs:comment annotations. The ontology was developed in Protégé by extracting all classes, subclasses, and their hierarchical relationships from the BPMN 2.0 specification, following the ontology engineering methodology proposed by Hildebrandt et al. [11]. We incorporated definitions from the BPMN 2.0 specification as rdfs:comment annotations to capture concept semantics. Using the BPMN 2.0 standard as the core ontology facilitates integration with domain-specific ontologies via ontology alignment techniques, enabling future applications that respect the modeling guidelines of individual public institutions. Fig. 2 depicts the class hierarchy of the BPMN ontology, which was included in the prompt as a component library (see Fig. 1, line 16) using RDF/Turtle syntax.

3 Quantifying the impact of ontology integration

To assess the impact of ontology integration, we compare process models generated by the **Process Copilot** using manually created ground-truth models as a reference. The ground-truth models are based on anonymized administrative processes. The LLM used is the *GPT-4.0* model. We selected ten processes of varying categories and complexity levels. For both quality and stability evaluations, we assess two dimensions of similarity: node similarity and structural similarity.

Node similarity. Node similarity is computed based on entity comparison. In a BPMN diagram, both activities and events are represented as nodes. For

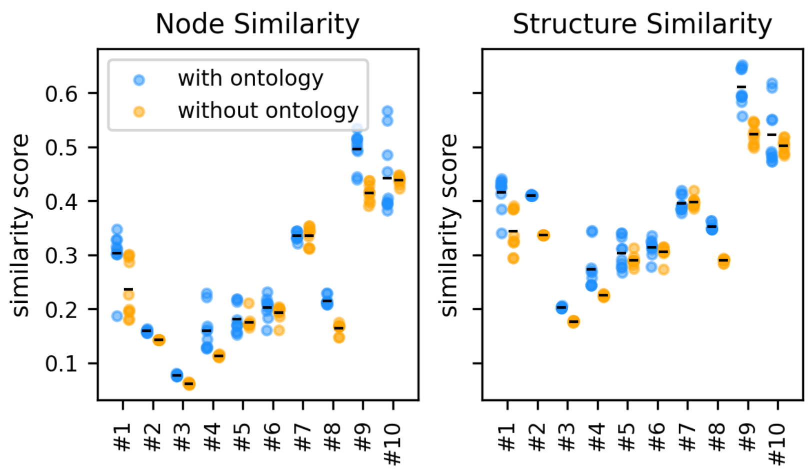

Fig. 3: **Including the ontology improves model performance.** The figure shows two side-by-side dot plots comparing node similarity and structural similarity between generated and ground-truth process models across ten different example processes (#1 to #10). Each dot represents the result of one run (10 runs per method per example), black lines indicate the mean similarity.

each node pair across two models, we compute the labels' string edit distance normalized by the maximum label length; the syntactic similarity. If the syntactic similarity exceeds a given threshold of 0.5, the nodes are considered equivalent. We use the threshold suggested in the corresponding paper [8]. The overall node similarity between two models is then calculated as the normalized sum of the syntactic similarities across matched node pairs.

Structural similarity. To assess structural similarity, the process models are converted into graphs, where nodes represent activities, events, and gateways, and are labeled accordingly. The structural similarity is then given by the normalized graph edit distance by the total number of nodes [8].

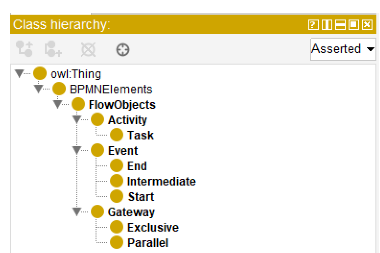

Fig. 2: Ontology class hierarchy.

Ontology integration improves performance. To evaluate performance, we ran each method ten times on each of the ten selected sets of ten process examples. The results are presented in Figure 3, which shows dot plots for both metrics across ten examples. The comparison reveals that ontology integration consistently results in higher average similarity scores. Based on a two-sided MannWhitney U-test, there is only weak evidence of a performance difference for node similarity ($p = 0.053$), whereas ontology integration significantly improves performance for structural similarity ($p = 0.007$).

Stability may be affected by increased complexity. The performance evaluation indicates a greater variance in model outputs when the ontology is incorporated. To quantify stability, we computed pairwise similarities between all outputs within each process individually. The mean node similarity was 0.835 with ontology and 0.866 without ontology usage respectively with a p-value of 0.00009 indicating statistically significant difference. Similarly, the mean structural similarity was 0.866 and 0.875 with and without ontology usage respectively, with a p-value of 0.004. In general, the similarities within each process remain high, indicating relatively strong output stability across all processes and both prompting strategies.

Discussion. Despite performance improvements, the absolute similarity scores remain moderate (0.3-0.6) reflecting the inherent variability and subjectivity in process modelling, where the lack of objectivity is a recognized challenge [2]. In addition, the ground-truth models include BPMN elements such as pools, message flows, catch events and text annotations, which are not represented in our component library or ontology, potentially penalizing structural mismatches outside the expressiveness of our generation system. However, stability is slightly lower with ontology integration, suggesting that the added semantic complexity introduces more variation in how the models are generated. This effect may stem from the expanded search space introduced by the ontology, especially in complex scenarios. Nevertheless, this increase in variance appears modest and does not outweigh the performance gains in model quality.

4 Summary & Outlook

This work highlights the potential of ontology-based knowledge integration to improve the semantic and structural quality of LLM-generated process models. While results show benefits, stability remains a concern.

Future work will explore the benefits of incorporating domain-specific ontologies. In particular, we aim to investigate algorithmic integration through constraints or validation to ensure consistent and accurate outputs. We also plan to assess the performance of open source LLMs for BPMN models. Additionally, we will evaluate Process Copilot using the PET dataset [3] to enable comparison with related tools. The PET dataset is a corpus of business process descriptions that are known to experts and considered relevant. This ensures comparability across different methods. Preliminary validation indicates that Process Copilot can already generate a wider range of BPMN element types than ProMoAI [15], supporting more complex process models, and we intend to extend its capabilities further by enabling BPMN subprocesses, frequently requested by our clients.

We will scale evaluations to larger datasets and may include hallucination metrics such as GraphEval [20] for more rigorous assessment. Techniques such as syntax correction with negative prompting, hallucination pruning [7], and agent-based restructuring might further enhance quality. Additional directions include interactive model editing, ontology-guided modeling hints, and integration of open-source models to support use in privacy-sensitive domains. Overall, unlocking the full potential of this approach will require deeper technical integration of knowledge and user-centric system design.

References

1. Bai, J., et al.: AutoSchemaKG: Autonomous knowledge graph construction through dynamic schema induction from web-scale corpora (2025). https://doi.org/10.48550/arXiv.2505.23628
2. Beerepoot, I., et al.: The biggest business process management problems to solve before we die. Computers in Industry **146**, 103837 (2023). https://doi.org/10.1016/j.compind.2022.103837
3. Bellan, P., van der Aa, H., Dragoni, M., Ghidini, C., Ponzetto, S.P.: PET: an annotated dataset for process extraction from natural language text tasks. In: Cabanillas, C., Garmann-Johnsen, N.F., Koschmider, A. (eds.) Business Process Management Workshops - BPM 2022 International Workshops, Münster, Germany, September 11-16, 2022, Revised Selected Papers. Lecture Notes in Business Information Processing, vol. 460, pp. 315–321. Springer (2022). https://doi.org/10.1007/978-3-031-25383-6_23
4. Buck, H.: Demographic change and the public sector (2007)
5. van Cauter, Z., Yakovets, N.: Ontology-guided knowledge graph construction from maintenance short texts. In: Proceedings of the 1st Workshop on Knowledge Graphs and Large Language Models (KaLLM 2024). pp. 75–84 (2024). https://doi.org/10.18653/v1/2024.kallm-1.8
6. Chalkidis, I., et al.: LEGAL-BERT: The muppets straight out of law school. In: Findings of the Association for Computational Linguistics: EMNLP 2020. pp. 2898–2904. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.findings-emnlp.261
7. Chen, H., et al.: SAC-KG: Exploiting large language models as skilled automatic constructors for domain knowledge graphs. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 4345–4360 (2024). https://doi.org/10.18653/v1/2024.acl-long.238
8. Dijkman, R., et al.: Similarity of business process models: Metrics and evaluation **36**(2), 498–516 (2011). https://doi.org/10.1016/j.is.2010.09.006
9. Duesterwald, E., et al.: FLOW-BENCH: Towards conversational generation of enterprise workflows (2025). https://doi.org/10.48550/arXiv.2505.11646
10. Grohs, M., et al.: Large language models can accomplish business process management tasks. In: International conference on business process management. pp. 453–465. Springer (2023). https://doi.org/10.1007/978-3-031-50974-2_34
11. Hildebrandt, C., et al.: Ontology building for cyberphysical systems: Application in the manufacturing domain. IEEE Transactions on Automation Science and Engineering **17**(3), 1266–1282 (2020). https://doi.org/10.1109/TASE.2020.2991777
12. Jiang, P., et al.: Text-augmented open knowledge graph completion via pre-trained language models. In: ACL (Findings). pp. 11161–11180 (2023). https://doi.org/10.18653/v1/2023.findings-acl.709
13. Kaplan, A., et al.: Combining knowledge graphs and large language models to ease knowledge access in software architecture research. In: SemTech4STLD 2024, co-located with the ESWC 2024. vol. 3697, pp. 76–82 (2024)

14. Kourani, H., et al.: Process modeling with large language models. In: International Conference on Business Process Modeling, Development and Support. vol. 511, pp. 229–244. Springer (2024). https://doi.org/10.1007/978-3-031-61007-3_18
15. Kourani, H., et al.: ProMoAI: Process modeling with generative AI (2024). https://doi.org/10.24963/ijcai.2024/1014
16. Köpke, J., Safan, A.: Introducing the bpmn-chatbot for efficient llm-based process modeling. In: Proceedings of the International Conference on Business Process Management (2024)
17. Mehenni, G., et al.: Ontology-constrained generation of domain-specific clinical summaries. In: International Conference on Knowledge Engineering and Knowledge Management. pp. 382–398. Springer (2025). https://doi.org/10.1007/978-3-031-77792-9_23
18. Mergel, I.: Digital transformation of the german state. Public Administration in Germany pp. 331–355 (2021). https://doi.org/10.1007/978-3-030-53697-8_19
19. Neuberger, J., et al.: A universal prompting strategy for extracting process model information from natural language text using large language models. In: International Conference on Conceptual Modeling. pp. 38–55. Springer (2024). https://doi.org/10.1007/978-3-031-75872-0_3
20. Sansford, H., et al.: GraphEval: A knowledge-graph based LLM hallucination evaluation framework (2024). https://doi.org/10.48550/arXiv.2407.10793
21. Singer, R.: An ontological analysis of business process modeling and execution (2019). https://doi.org/10.48550/arXiv.1905.00499
22. Weerakkody, V., Janssen, M., El-Haddadeh, R.: The resurgence of business process re-engineering in public sector transformation efforts: exploring the systemic challenges and unintended consequences. Information Systems and e-Business Management **19**(3), 993–1014 (2021). https://doi.org/10.1007/s10257-021-00527-2
23. Yao, L., Mao, C., Luo, Y.: KG-BERT: BERT for knowledge graph completion (2019). https://doi.org/10.48550/arXiv.1909.03193