LLM4BPMNGen: A Tool for BPMN Generation with LLMs

Ana Costa[1,*], Alena Wimmer[1] and Luise Pufahl[1]

[1]Technical University of Munich, School of Computation, Information and Technology, Heilbronn, Germany

Abstract

Conceptual modeling techniques serve as a foundation for reliable process representations across various domains. In business process modeling, challenges such as incomplete, outdated, or entirely absent process documentation lead to reliance on extensive communication between domain experts and modelers. Business Process Modeling and Notation (BPMN) has become a standard in addressing these challenges, yet generating accurate BPMN models from textual descriptions remains difficult. This paper introduces LLM4BPMNGen, a tool leveraging LLMs to directly transform textual process descriptions into BPMN XML models, and covering a large set of BPMN elements. Preliminary analysis indicates high usability, although minor layout adjustments were suggested. Future work aims to address performance issues of the tool and enhance flexibility regarding LLM providers.

Keywords

Business Process Generation Tool, Business Process Modeling and Notation, Large Language Models

1. Introduction

A significant challenge within the conceptual modeling of processes remains in practice. Industry often lacks process documentation, and when documentation exists, it is usually outdated or incomplete [1, 2]. In addition to that, the experts with operational knowledge are not the same individuals who are modeling them [3]. Therefore, there is a constant necessity for knowledge exchange between domain experts and process modelers, which not only involves several rounds of interviews, meetings, surveys, or workshops [4], but also is prone to errors, ambiguities, misunderstandings, and loss of information.

Business Process Modeling and Notation (BPMN) has emerged as the industry standard for process modeling due to its ability to comprehensively communicate complex execution logic [5]. Consequently, researchers have been addressing the task of extracting process information from textual descriptions and generating models for decades [6]. The fact is, this problem has not yet been solved, and we have doubts about whether it's ever going to be, since both the language and the model are abstract representations of reality that exclude much of the world's detail [7, 8]. The purpose is, however, to reduce the complexity of understanding a domain expert by visualizing graphically what was performed as accurately as possible with the expert's intentions. The advancements in LLMs have substantially improved performance in handling complex and context-dependent textual descriptions, and researchers have not yet fully understood their capabilities for generating BPMN models [9, 10].

The necessity for robust, BPMN-focused solutions is evident across a variety of real-world contexts. In higher education, for example, students learning BPMN often struggle to validate the syntactic and semantic correctness of their models, leading to misunderstandings of the notation and restricted learning [11]. A tool that translates textual descriptions directly into BPMN diagrams provides immediate feedback, thereby supporting the educational process. Within industry, clear and accessible process modeling is essential for efficient communication between stakeholders [12]. For example, in sectors such as automotive manufacturing, operational changes require rapid alignment between managers and IT analysts, and the use of non-standard notations leads to delays and miscommunications [13]. By adopting a tool that directly translates the manager's explanations into BPMN models, it simplifies communication between business stakeholders and IT teams. In highly regulated environments, such as pharmaceuticals, process analysts must model complex procedures based on expert input, which is often

ER2025: Companion Proceedings of the 44th International Conference on Conceptual Modeling: Industrial Track, ER Forum, 8th SCME, Doctoral Consortium, Tutorials, Project Exhibitions, Posters and Demos, October 20-23, 2025, Poitiers, France

*Corresponding author.

a.costa@tum.de (A. Costa); luise.pufahl@tum.de (L. Pufahl)

provided in unstructured, context-dependent language [14]. Traditional NLP-based methods frequently fall short in extracting complex information, and the integration of advanced LLM techniques enables the generation of BPMN models that capture complex language dependencies.

We address these challenges and real-world scenarios by proposing a tool that performs all tasks from information extraction to BPMN generation fully with LLM techniques on prompt engineering, covering an extended set of BPMN elements, and aiming at producing syntactically and semantically correct BPMN XML files. We innovate in comparison to other tools by broadening the set of supported BPMN elements and by proposing a direct transformation approach from extraction to generation, completely leveraged by LLMs.

In the following sections, we present related work mostly on recent approaches for BPMN generation (section 2), the tool description with preliminary analysis (section 3), and conclusion and future work (section 4).

2. Related Work

Recent approaches for BPMN generation use LLMs for either one or both information extraction and BPMN model generation [15, 16, 17]. This direct or two-step transformation approach [18] maps a process model via an ad-hoc pipeline [19], or uses an intermediate representation to create the process model and capture the essential elements of a process [20, 21].

Practically, LLMs are suitable for tasks like process modeling that require the generation of outputs from process description due to their capability to process and interpret natural language [22]. The BPMN Sketch Miner uses a rule-based approach to define the process elements and applies an algorithm to generate diagram metadata with LLMs [9]. [10] apply two rule-based steps to transform their graph-based concept map into the BPMN diagram. `ProMoAI` transforms an intermediate representation into the corresponding diagram using the `pm4py` library and LLMs [23]. [24] use a rule-based approach to transform their intermediate representation into a corresponding BPMN, and [25] use the same approach to map patterns of trees to BPMN diagrams. [16] demonstrate the suitability for extracted information for model generation with prompting strategies on eight different LLMs. [26] present an interactive tool for generation with LLMs that takes into consideration the number of tokens of the prompting strategies and output formats. Finally, Nala2BPMN uses an LLM-based module for information extraction and an algorithmic module for generation [17].

In summary, the number of approaches for BPMN generation increased drastically since the advent of LLMs, which reconfirms its suitability for the task. However, none of the presented approaches supports advanced BPMN elements such as sub-processes, event-based gateways, black-box pools, and other more complex elements within the notation. Our tool supports an advanced set of elements and models complex interdependencies within the processes, with a direct transformation approach based on a three-step prompting technique, and provides output in an Extensible Markup Language (XML) format that is directly importable into commercial and academic tools.

3. LLM4BPMNGen Tool

The tool [1] consists of three different modules: (1) Input Processing, (2) Process Generation, and (3) Visualization (Figure 1). The user interacts with the application via the `Streamlit` web interface, providing the API key and selecting from three input modes: audio upload, voice recording, or text input. The input processing includes converting spoken descriptions to text using an LLM, and ensuring that the input is passed to the process generation module in a unified format.

Process Generation is the core logic of the tool since it transforms process descriptions into XML models using prompt-based interactions with the LLM. It includes steps for the initial generation, iterative improvement, and completion of the BPMN diagram. After the input is received, (1) it

[1] https://bpmn-generation.streamlit.app/
https://youtu.be/G8sORt-D5uE?si=1porrvGqYkiP7yi2

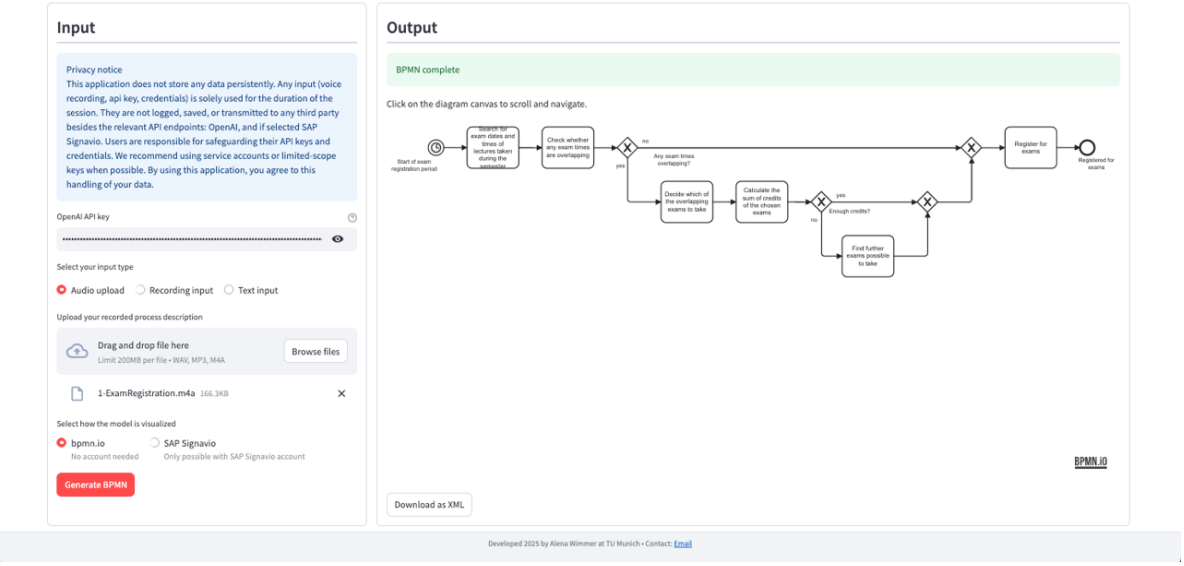

Figure 1: LLM4BPMNGen after the generation of a process.

transforms speech into text if applicable, and (2) sends a two-shot chain-of-thought prompt to an LLM, receiving the first version of the XML (BPMN generation). Then, (3) it sends the first version, the description, and a refinement prompt to the LLM, expecting a refined version of the XML (BPMN refinement). Finally, (4) it ensures that the visual elements of the XML match the logical process elements (completeness check). The **Visualization** module handles the display of the final BPMN model supporting two options, both leveraged with third-party libraries. Either the XML is rendered client-side via an embedded viewer through an HTML component, or it is retrieved as a rendered image of the model via the API of an external provider. In both cases, the visualization can be downloaded as an XML file.

The tool utilizes the OpenAI API with the `whisper-1` model used to transcribe the audio and the `o1` GPT model used for three chained calls. All requests to the API are authenticated via a user-provided API key. Optionally, an external provider API can be used to visualize the generated model. Regarding credential handling and privacy, the user enters the API key and credentials at runtime, and they are never persistently stored. All data handling takes place within the session, and no data is transmitted to third parties beyond the called APIs. Additionally, a privacy notice is displayed in the interface, emphasizing the user's responsibility for sensitive data. Furthermore, the tool handles errors regarding possible authentication problems, upload or export, and possible visualization errors, presenting real-time feedback to the user. In case of an API authentication error, the user is asked to enter a valid API key. If there is a visualization error from the external provider, the system redirects to a standard visualization. In both cases, and in case of unexpected errors, the error details are displayed to the user, who is asked to retry.

Four guiding principles were considered for the conceptual design. Modularity since the components were encapsulated with clear responsibilities, flexibility since multiple input and output modalities are supported, fail-safety since the system includes fallback mechanisms for switching visualization mode, error propagation, and isolate failure sources, and extensibility since the design supports future extensions and the use of different LLM providers.

3.1. Preliminary analysis

A preliminary analysis of the tool was performed to collect expert feedback on both the usability of the developed tool and the quality of the generated models. The six participants received a brief explanation of the tool's functionality along with a link to access the system and the anonymous survey. They were asked to try out the tool independently using process descriptions of their choice, and then complete

the anonymous survey based on their experience. Besides questions such as the participant's experience with process modeling and software usage context, the System Usability Scale (SUS) questions were asked [27], as well as custom questions regarding model alignment, syntactic correctness, readability, and required adjustments.

The averaged SUS value of 84.58 indicates good usability [28]. In general, participants strongly agreed that the system is easy to learn and use, and the experts were overall very satisfied with the user interface of the artifact. All experts agreed that the models reflect the content of their input, which indicates that the artifact is able to extract the process information well. However, only 67% of the experts felt that their generated models correctly followed the BPMN syntax and were clear and easy to understand. Most experts agree that the model can be used as a basis for further process modeling or automation, with only minor or moderate adjustments required, and no expert stated that major adjustments would be necessary or that the model would have to be rebuilt from scratch. Key issues identified by the experts include minor layout issues in the models, long processing time, and different modeling choices.

4. Conclusion and Future Work

We present a tool for BPMN model generation that performs all tasks from information extraction to model generation with LLMs. LLM4BPMNGen considers a large set of BPMN elements that are not yet addressed in current literature. Experts in the preliminary analysis provide improvement suggestions to the tool, such as layout issues, long processing time, and different modeling choices that should be addressed as future research. LLM4BPMNGen can be widely adopted in different scenarios, and we highly recommend the use of the tool, especially for educational purposes. As limitations, the reliance on external LLM API and computational resources limits the accessibility when providers are not available. We recommend therefore, extensions to accommodate other providers in the future.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] H. Van der Aa, J. Carmona Vargas, H. Leopold, J. Mendling, L. Padró, Challenges and opportunities of applying natural language processing in business process management, in: COLING 2018, Association for Computational Linguistics, 2018, pp. 2791–2801.

[2] W. Van Woensel, S. Motie, Nlp4pbm: a systematic review on process extraction using natural language processing with rule-based, machine and deep learning methods, Enterprise Information Systems (2024) 2417404.

[3] J. Herbst, D. Karagiannis, An inductive approach to the acquisition and adaptation of workflow models, in: Proceedings of the IJCAI, volume 99, Citeseer, 1999, pp. 52–57.

[4] L. B. Hvatum, Requirements elicitation with business process modeling, in: Proceedings of the 21st Conference on Pattern Languages of Programs, 2014, pp. 1–9.

[5] J. Recker, Opportunities and constraints: the current struggle with bpmn, Business Process Management Journal 16 (2010) 181–201.

[6] A. Wimmer, A. Costa, L. Pufahl, Natural language processing for bpmn model generation with llms: A systematic literature review, in: Business Process Management Workshops 2025, Springer, 2025, p. accepted.

[7] B. Curtis, M. I. Kellner, J. Over, Process modeling, Commun. ACM 35 (1992) 75–90.

[8] A. Schaff, Language and reality, Diogenes 13 (1965) 147–167.

[9] A. Ivanchikj, S. Serbout, C. Pautasso, Live process modeling with the bpmn sketch miner, Software and systems modeling 21 (2022) 1877–1906.

[10] R. Sonbol, G. Rebdawi, N. Ghneim, A machine translation like approach to generate business process model from textual description, SN Computer Science 4 (2023) 291.
[11] I. Maslov, Towards empirically validated process modelling education using a bpmn formalism, in: R. Guizzardi, J. Ralyté, X. Franch (Eds.), Research Challenges in Information Science, Springer International Publishing, Cham, 2022, pp. 803–810.
[12] J. Recker, M. Indulska, P. Green, Extending representational analysis: Bpmn user and developer perspectives, in: G. Alonso, P. Dadam, M. Rosemann (Eds.), Business Process Management, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 384–399.
[13] E. Knauss, P. Pelliccione, R. Heldal, M. Ågren, S. Hellman, D. Maniette, Continuous integration beyond the team: A tooling perspective on challenges in the automotive industry, ESEM '16, Association for Computing Machinery, New York, NY, USA, 2016.
[14] G. M. Troup, C. Georgakis, Process systems engineering tools in the pharmaceutical industry, Computers Chemical Engineering 51 (2013) 157–171. CPC VIII.
[15] R. Sonbol, G. Rebdawi, N. Ghneim, A machine translation like approach to generate business process model from textual description, SN Computer Science 4 (2023) 291.
[16] J. Neuberger, L. Ackermann, H. van der Aa, S. Jablonski, A universal prompting strategy for extracting process model information from natural language text using large language models, in: W. Maass, H. Han, H. Yasar, N. Multari (Eds.), Conceptual Modeling, Springer Nature Switzerland, Cham, 2025, pp. 38–55.
[17] A. Nour Eldin, N. Assy, O. Anesini, B. Dalmas, W. Gaaloul, Nala2bpmn: Automating bpmn model generation with large language models, in: M. Comuzzi, D. Grigori, M. Sellami, Z. Zhou (Eds.), Cooperative Information Systems, Springer Nature Switzerland, Cham, 2025, pp. 398–404.
[18] P. Bellan, M. Dragoni, C. Ghidini, H. van der Aa, S. P. Ponzetto, Process extraction from text: Benchmarking the state of the art and paving the way for future challenges, arXiv preprint arXiv:2110.03754 (2021).
[19] H. van der Aa, C. Di Ciccio, H. Leopold, H. A. Reijers, Extracting declarative process models from natural language, in: Advanced Information Systems Engineering: 31st International Conference, CAiSE 2019, Rome, Italy, June 3–7, 2019, Proceedings 31, Springer, 2019, pp. 365–382.
[20] F. Friedrich, J. Mendling, F. Puhlmann, Process model generation from natural language text, in: Advanced Information Systems Engineering: 23rd International Conference, CAiSE 2011, London, UK, June 20-24, 2011. Proceedings 23, Springer, 2011, pp. 482–496.
[21] K. Honkisz, K. Kluza, P. Wiśniewski, A concept for generating business process models from natural language description, in: Knowledge Science, Engineering and Management: 11th International Conference, KSEM 2018, Changchun, China, August 17–19, 2018, Proceedings, Part I 11, Springer, 2018, pp. 91–103.
[22] P. Bellan, M. Dragoni, C. Ghidini, Process knowledge extraction and knowledge graph construction through prompting: A quantitative analysis, in: Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing, 2024, pp. 1634–1641.
[23] H. Kourani, A. Berti, D. Schuster, W. M. van der Aalst, Promoai: Process modeling with generative ai, arXiv preprint arXiv:2403.04327 (2024).
[24] N. Daclin, S. Mallek-Daclin, G. Zacharewicz, Generative ai for business model generation (gai4bm): from textual description to business process model, in: the 10th International Food Operations and Processing Simulation Workshop, CAL-TEK srl, 2024.
[25] Q. Nivon, G. Salaün, Automated generation of bpmn processes from textual requirements, in: International Conference on Service-Oriented Computing, Springer, 2024, pp. 185–201.
[26] J. Köpke, A. Safan, Efficient llm-based conversational process modeling, in: K. Gdowska, M. T. Gómez-López, J.-R. Rehse (Eds.), Business Process Management Workshops, Springer Nature Switzerland, Cham, 2025, pp. 259–270.
[27] J. Brooke, et al., Sus-a quick and dirty usability scale, Usability evaluation in industry 189 (1996).
[28] J. Brooke, Sus: a retrospective, J. Usability Studies 8 (2013) 29–40.