Nisaba: Towards Generating Natural Language Description of Multi-Perspective Declarative Process Models

Wesley da Silva Santos[1], Juliana Rezende Coutinho[2], Fernanda Baião[2], Georges Miranda Spyrides[1], and Hélio Côrtes Vieira Lopes[1]

[1] Department of Informatics, PUC-Rio, Rio de Janeiro, Brazil
{wsantos,gspyrides,lopes}@inf.puc-rio.br
[2] Department of Industrial Engineering, PUC-Rio, Rio de Janeiro, Brazil
{julianarezendecoutinho@aluno.,fbaiao@}puc-rio.br

Abstract. Declarative approaches in Business Process Management (BPM) offer flexibility but pose comprehension challenges, especially for non-technical stakeholders. Process model descriptions are crucial in bridging the gap between technical representations and business users, enhancing model validation and communication. This work presents a step towards addressing these challenges through Nisaba, a prototype tool aimed at generating comprehensible natural language descriptions of Multi-Perspective Declarative (MP-Declare) process models. Leveraging Large Language Models (LLMs) and prompt engineering techniques (PE), Nisaba explores overcoming limitations of traditional Natural Language Generation (NLG) methods. The tool integrates a Python-based interface with generative AI modules, applying computational linguistics to process MP-Declare models and produce tailored descriptions. This research serves as a primer for making declarative BPM more accessible, potentially enhancing process design, analysis, and implementation in dynamic business environments.

Key words: declarative process management, process models, mp-declare, synthetically generated data, large language models

1 Introduction

The field of Business Process Management (BPM) has evolved towards Declarative Business Process Management (DBPM), emphasizing rules and constraints over predefined sequential steps. Foundational work by Deutch & Milo [1] and van der Aalst et al. [2] established this shift, with the former integrating declarative query languages and the latter introducing the Declare framework. The evolution to Multi-Perspective Declare (MP-Declare) [3] further expanded DBPM's capabilities. However, this transition presents significant challenges, as the shift from procedural to declarative models requires new technical skills and a fundamental change in conceptualizing process models ([4]). A key challenge is the tendency to impose sequential narratives on non-linear, declarative models [5], complicating understanding and maintenance. Process model descriptions play a crucial role in facilitating comprehension, with research by Haisjackl et al. [6] highlighting the importance of effective labeling and representation methods.

Recent advancements in generative artificial intelligence (GAI), particularly Large Language Models (LLMs), offer promising solutions to these challenges. The potential of LLMs in BPM has been posed by [7] and explored by researchers such as Busch et al. [8] and Vidgof et al. [9]. Building on this foundation, our research leverages LLMs to enhance the understandability of declarative business process models, focusing on developing a tool for generating natural language descriptions of MP-Declare models. This paper presents our approach to generate accurate, context-aware, and easily understandable descriptions. By addressing the cognitive challenges associated with interpreting declarative models, we aim to bridge the gap between their flexible nature and diverse stakeholders, contributing to making declarative BPM more accessible and practical.

2 Process Model Understandability and Description Generation

Research on process model understandability has increasingly focused on the role of process descriptions. Engiel et al. [10] found that presenting a process model as a written use case description followed by its BPMN diagram was most effective, especially for readers without formal training. This approach highlights the importance of textual descriptions in bridging the gap between expert modelers and lay audiences. The literature strongly supports the significance of process descriptions [6, 7] in enhancing model understandability, whether through careful labeling, abstract textual representations, or combining textual and graphical elements.

The field of generating descriptions for process models has evolved significantly, addressing challenges specific to declarative models and multilingual contexts. Ackermann et al. [12] proposed a method for generating natural language texts from declarative process models, focusing on content determination and text structuring to improve readability. Rodrigues et al. [13] introduced BPMN2TEXT, a framework generating multilingual texts from BPMN models, aimed at helping domain specialists without modeling skills. While these approaches demonstrate growing sophistication, they rely on classical natural language generation techniques [14] with limitations in context understanding, evaluation consistency, and adaptability to new domains. These limitations include producing grammatically correct but unnatural text and requiring labor-intensive handcrafted linguistic features and rules for retraining in new domains.

3 Tool and Employed Techniques

The tool for generating natural language descriptions of MP-Declare process models utilizes a Python Notebook with a Gradio-enabled GUI for user inter-

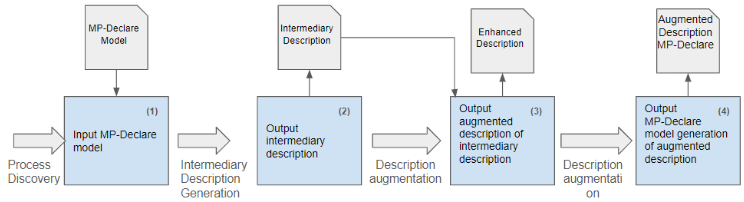

Fig. 1. The generation flow of Nisaba.

action. This system comprises two main components: the Python Notebook, which serves as the primary interface for uploading models and viewing generated descriptions, and the Generative AI Modules, which leverage OpenAI's `gpt-4o-2024-05-13` model with a low temperature setting (0.2) to enhance reproducibility and replicability [15, 16]. The tool's flow is illustrated in Figure 1, with the Python Notebook handling user interactions and pre-processing, while the Generative AI component, consisting of four modules integrating PE techniques, processes the MP-Declare models to produce context-aware, continuous descriptions. This approach aims to address the specific challenges of declarative process models, ensuring that the generated descriptions accurately reflect the model's nuances and enhance overall understandability.

The goal of step (1) is to process the MP-Declare model to validate and generate the prompt for creating intermediary descriptions. Users can attach a ".decl" file or paste the model lines, which are then parsed by the Declare4Py framework, validated against its patterns, and used to generate a Declare4Py object. The model is then structured into a prompt, incorporating each part of the model lines (activities, binds, attributes, and constraints). In step (2), this prompt serves as input to generate intermediary descriptions using prompt engineering (PE) and function calling. Function calling enhances AI assistants by enabling integration with external tools and systems [17]. We created four functions to fetch, structure, and output the constructs as JSON objects for activities, binds, attributes, and constraints. This structured output generates granular descriptions of the constructs, adding semantics for future validation and quality generation. The model is infused with MP-Declare semantics, instructing it to output with the desired semantic structure.

In step (3), we generate the desired description for the end user by utilizing the intermediary descriptions from step (2) and additional PE techniques to structure it as continuous text. This approach is based on Leopold et al.'s [11] method for generating imperative process descriptions, which has shown good results in understandability. We structure the text following the hierarchy of MP-Declare constraints proposed by Di Ciccio and Mecella [18], presenting information from the most generalized to the most specific constraints, thus reducing cognitive load [19] and enhancing understandability. The final step (4) involves extracting the MP-Declare model from the generated description, performing a reverse generation to compare the extracted model with the original in terms of complexity and similarities. This step uses the same function as in step (2) to structure the main concepts of MP-Declare, employing an instruction prompt to capture these elements from the continuous text.

4 Evaluation and Conclusions

Table 1. Evaluation Results

Model ID	**Questions**				
	Q1	***Q2***	***Q3***	***Q4***	***Q5***
1	9	9	8	9	8
2	9	9	8	9	9
3	8	8	7	8	9
4	8	9	7	8	8
5	8	9	8	8	7

To evaluate our proposal, we employed a set of five process models varying in domain and complexity[1]. Our assessment was twofold: we examined understandability both qualitatively and empirically, and applied an established metric from literature to assess model completeness. Given the diverse ways to operationalize model understandability and the existing gap in literature for a comprehensive evaluation framework [20], we focused on the effectiveness dimension. We assumed that our PE-enhanced descriptions of MP-Declare constraints would elucidate process nuances, thereby facilitating model understanding.

We employed a combination of open-ended and closed-ended questions to compare final descriptions with intermediary descriptions. Given the challenges in evaluating continuous text outputs, particularly for open-ended questions, we utilized LLMs as judges. This allows for customized assessments while accommodating open-ended responses, with research showing a strong correlation between LLM-generated scores and human feedback [21]. Our evaluation consisted of five questions[2], rated on a scale of 1 to 10, assessing various aspects of the descriptions' quality. These questions covered the adherence to guidelines, capture of essential MP-Declare model elements, understandability for non-technical audiences, balance between detail and clarity, and effectiveness in explaining data attributes and resource integration. Respondents were required to provide both a rating and a justification for each question, enabling quantitative assessment and qualitative insights into the descriptions' understandability and effectiveness.

Table 1 shows the evaluation results for Nisaba's descriptions of five MP-Declare process models. Scores ranged from 7 to 9 out of 10 across all criteria, indicating high overall performance. The tool excelled in capturing essential elements (Q2) and adhering to guidelines (Q1), with scores mostly in the 8-9 range.

[1] Available at: `https://github.com/santos-wesley/Nisaba/tree/main/Models`

[2] Each complete question are inside the models folder: `https://github.com/santos-wesley/Nisaba/blob/main/Models/Questions.txt`

Balance between detail and clarity (Q4) was well-maintained. Data and resource integration explanations (Q5) showed some variability (7-9). Non-technical understanding (Q3) received the lowest scores (7-8), suggesting room for improvement. Overall, Nisaba effectively generates comprehensive and understandable descriptions, accurately representing model elements and following guidelines.

We assessed description completeness using metrics proposed by Abbad-Andaloussi et al. [22]: size, density, separability, and constraint variability. Table 2 compares original models to those reconstructed from generated descriptions. Models 1, 2, and 5 maintained exact size, while Models 3 and 4 had slight reductions. Density remained identical except for a small increase in Model 5. Separability was preserved for Models 1, 2, and 5, with minor differences in Models 3 and 4. Constraint variability matched perfectly for Models 1, 2, and 5, remained close for Model 3, and dropped to zero for Model 4. Overall, the generated descriptions largely preserved the original models' structural properties, indicating high completeness in the natural language generation process. Evaluation of Nisaba using five diverse process models showed its effectiveness in generating high-quality, understandable descriptions adhering to key criteria. The LLM-based approach offers advantages over traditional techniques, including context-awareness, domain adaptability, and more natural text. Challenges remain with very large or complex models. Future work will explore advanced LLMs, additional PE techniques, and user studies to improve quality and assess real-world impact. Nisaba advances the accessibility of declarative process models, promising enhanced collaboration in dynamic business environments.

Table 2. Comparison of Models Based on Different Metrics

Metric	Model 1		Model 2		Model 3		Model 4		Model 5	
	Original	Description	Original	Description	Original	Description	Original	Description	Original	Description
Size	7	7	19	19	30	28	67	28	30	30
Density	0.75	0.75	0.888889	0.888889	0.8	0.8	3.5	0.8	0.928571	0.928571
Separability	0.142857	0.142857	0.0526316	0.0526316	0.166667	0.25	0.0149254	0.178571	0.0666667	0.0666667
Constraint Variability	0	0	0	0	0.773706	0.861353	0.276195	0	0	0

References

1. D. Deutch and T. Milo, "A quest for beauty and wealth (or, business processes for database researchers)," in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, 2011.
2. W. M. P. van der Aalst, M. Pesic, and H. Schonenberg, "Declarative workflows: Balancing between flexibility and support" *Computer Science - R&D*, 2009.
3. A. Burattin, F. M. Maggi, and A. Sperduti, "Conformance checking based on multi-perspective declarative process models" *Expert Systems with Applications*, 2016.
4. D. Fahland, D. Lübke, J. Mendling, H. Reijers, B. Weber, M. Weidlich, and S. Zugal, "Declarative versus Imperative Process Modeling Languages: The Issue of Understandability" in *Enterprise, Business-Process and Information Systems Modeling*, Springer, 2009.

5. C. Haisjackl, S. Zugal, P. Soffer, I. Hadar, M. Reichert, J. Pinggera, and B. Weber, “Making Sense of Declarative Process Models: Common Strategies and Typical Pitfalls” in *Advanced Information Systems Engineering Workshops*, Springer, 2013.
6. C. Haisjackl and S. Zugal, “Investigating Differences between Graphical and Textual Declarative Process Models” in *Advanced Information Systems Engineering Workshops*, L. Iliadis, M. Papazoglou, and K. Pohl, Eds., Springer International Publishing, Cham, 2014.
7. S. Feuerriegel, J. Hartmann, C. Janiesch, and P. Zschech, “Generative AI” *Business & Information Systems Engineering*, 2024.
8. K. Busch, A. Rochlitzer, D. Sola, and H. Leopold, “Just tell me: Prompt engineering in business process management” in *International Conference on Business Process Modeling, Development and Support*, Springer, 2023.
9. M. Vidgof, S. Bachhofner, and J. Mendling, “Large Language Models for Business Process Management: Opportunities and Challenges” in *BPM*, Springer, 2023.
10. P. Engiel, R. Araujo, and C. Cappelli, “Designing public service process models for understandability,” *Electronic Journal of e-Government*, 2014.
11. H. Leopold, J. Mendling, and A. Polyvyanyy, “Supporting process model validation through natural language generation” *IEEE Trans. on Software Engineering*, 2014.
12. L. Ackermann, S. Schönig, M. Zeising, and S. Jablonski, “Natural language generation for declarative process models” in *Enterprise and Organizational Modeling and Simulation: 11th International Workshop, EOMAS 2015, Held at CAiSE 2015, Stockholm, Sweden, June 8-9, 2015, Selected Papers 11*, Springer, 2015.
13. R. A. Rodrigues, L. G. Azevedo, and K. Revoredo, “BPMN2TEXT: A Language-Independent Framework to Generate Natural Language Text from BPMN models” *iSys - Brazilian Journal of Information Systems*, 2016.
14. E. Reiter and R. Dale, “Building applied natural language generation systems,” *Natural Language Engineering*, 1997.
15. T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, “Language Models are Few-Shot Learners” in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., Curran Associates, Inc., 2020.
16. P. Ivie and D. Thain, “Reproducibility in scientific computing” *ACM CSUR*, 2018.
17. OpenAI, “Function Calling Guide,” 2024. [Online]. Available: `https://platform.openai.com/docs/guides/function-calling`. Accessed: 2024-07-03.
18. C. Di Ciccio and M. Mecella, “A two-step fast algorithm for the automated discovery of declarative workflows” in *2013 IEEE Symposium on Computational Intelligence and Data Mining*, IEEE, 2013.
19. J. Sweller, J. J. G. Van Merrienboer, and F. G. W. C. Paas, “Cognitive architecture and instructional design” *Educational Psychology Review*, 1998.
20. C. Houy, P. Fettke, and P. Loos, “Understanding understandability of conceptual models–what are we actually talking about?” in *Conceptual Modeling: 31st International Conference ER 2012, Florence, Italy, Oct, 2012. Proceedings*, Springer, 2012.
21. L. Zheng, W. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, et al., “Judging LLM-as-a-judge with MT-Bench and Chatbot Arena” *Advances in Neural Information Processing Systems*, 2024.
22. A. Abbad-Andaloussi, A. Burattin, T. Slaats, E. Kindler, and B. Weber, “Complexity in declarative process models: Metrics and multi-modal assessment of cognitive load” *Expert Systems with Applications*, 2023.