L. Pufahl, J.-R. Rehse (Eds.): EMISA 2025,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2025 105

Petri Net of Thoughts: A Structure-Enhanced Prompting Approach for Process-Aware Artificial Intelligence

Aleksandar Gavric [1], Dominik Bork [1], and Henderik Proper [1]

Abstract: Prompt engineering, with techniques such as Chain/Tree/Graph-of-Thoughts, has emerged as a key method to enhance the reasoning capabilities of Large Language Models (LLMs) by guiding them through sequential and multi-path problem-solving approaches. In this paper, we introduce *Petri Net of Thoughts (PNoT)*, a novel structure-enhanced prompting paradigm that leverages process discovery techniques to guide the reasoning process of an LLM during the generation of a Petri net model. PNoT represents the sequence, concurrency, and decision-making inherent in complex tasks. PNoT aims to improve both efficiency and interpretability in structured LLM-based reasoning tasks within the business process management domain.

Keywords: Prompt Engineering, Structured Reasoning, Business Process Management, Large Language Models

1 Introduction

Prompt engineering has emerged as a critical technique for enhancing the reasoning capabilities of Large Language Models (LLMs). Techniques such as *Chain-of-Thought (CoT)* prompting guide LLMs in breaking down problems into sequential intermediate steps [We23]. More recent paradigms, such as *Tree-of-Thoughts (ToT)* [Ya23b] and *Graph-of-Thoughts (GoT)* [Be24], expand CoT by allowing branching and richer dependency structures in the reasoning process. In parallel, the field of *process mining* has developed formal methods for modeling and analyzing processes using *Petri nets* [Re85, VDA12]. Petri nets consist of *places* and *transitions* connected by arcs, providing a mathematically rigorous framework for modeling concurrency, causality, and synchronization [Mu89]. Process discovery techniques, such as the Alpha Miner algorithm, analyze event logs to infer a Petri net model that represents the underlying process [vdAWM04]. Despite the advancements in LLM prompting strategies within the context of business processes [Fi24, CBS23], there remains a gap in methodologies that explicitly integrate formal process modeling techniques to enhance LLM reasoning—enabling traceable, logically consistent, and domain-aware decision-making.

In this paper, we propose *Petri Net of Thoughts (PNoT)*—a prompting approach that marries process mining with LLM prompting. PNoT treats the reasoning process as an event log, uses process discovery to derive a Petri net, and then employs this net to guide the LLM

1 Business Informatics, TU Wien, Erzherzog-Johann-Platz 1, 1040, Vienna, Austria,
aleksandar.gavric@tuwien.ac.at, https://orcid.org/0009-0005-1243-7722;
dominik.bork@tuwien.ac.at, https://orcid.org/0000-0001-8259-2297;
henderik.proper@tuwien.ac.at, https://orcid.org/0000-0002-7318-2496

DOI: https://doi.org/10.18420/EMISA2025-15

during inference. This approach aims to capture the business logic of a process in a formal sense which replicates how decisions are made by participants in real-world scenarios. Unlike CoT, ToT, and GoT, where the reasoning schedule might be abstract or loosely structured, PNoT uses well-defined (discovered) places and transitions to simplify and focus decision-making. This concrete structure aims to help isolate moments of sequential and concurrent reasoning, allowing for a more organized and contextual aware reasoning. PNoT aims to ensure that an LLM's reasoning follows a process structure modeled on evidence (via process discovery) or confidence (via domain experts) levels, thereby enhancing the transparency and verifiability of its intermediate steps. By mapping the internal states of the LLM onto Petri net components, PNoT aims to enable a systematic exploration of possible reasoning paths. This formalism facilitates error analysis and debugging by allowing practitioners to trace back through the net's transitions, thereby illuminating where and why an inference may have gone astray. Moreover, the integration of process mining techniques into LLM prompting opens the door for adaptive reasoning, where the system can dynamically adjust its strategy based on feedback extracted from the event log, ultimately enhancing both robustness and performance on complex, multi-step tasks. In light of these advantages, our primary research question (RQ) is: **How can the structural and behavioral properties of Petri nets be used to design a prompting strategy that formalizes and guides LLM reasoning in the context of business processes?** In the remainder, Section 2 positions our work compared to related works. Section 3 details the PNoT paradigm and a running example. Section 4 concludes with future directions.

2 Related Work

Over the past few years, a variety of prompting paradigms have been introduced to enhance the reasoning capabilities of LLMs. Early work such as *Chain-of-Thought (CoT)* prompting [We23] laid the groundwork by guiding LLMs to generate intermediate reasoning steps. Building on this idea, the *Tree-of-Thought (ToT)* framework was proposed in multiple forms. For instance, one variant [Ya23b] focuses on a systematic tree structure that encapsulates reasoning hierarchies. Subsequent methods have further diversified this line of inquiry. The *Tree of Uncertain Thought* [MX23] and *Tree-of-Mixed-Thought* [Hu23] approaches tackle ambiguity and the integration of heterogeneous reasoning pathways, respectively. Similarly, *Skeleton-of-Thought* [Ni24] strips down the reasoning process to its essential components, aiming for a minimalist yet effective structure. More recently, researchers have introduced the *Graph of Thought* (GoT) method [Be24], which represents reasoning as interconnected nodes and edges to better capture complex dependencies, while the *Everything of Thoughts* (XoT) approach [Di24] proposes an even more comprehensive framework that leverages pre-trained reinforcement learning and Monte Carlo Tree Search. The *Hypergraph-of-Thought* model [Ya23a] generalizes these ideas by accounting for higher-order relationships between reasoning elements. Parallel to these innovations, techniques emphasizing *Thought Propagation* [YHY24] and *Socratic Questioning* [Qi23] have been developed to foster iterative refinement of reasoning. These methods encourage the LLM to continuously reexamine its assumptions and validate each inference step, thereby improving both transparency and robustness.

In contrast to the aforementioned approaches that primarily focus on the structure of the reasoning process itself, our work introduces the *Petri Net of Thoughts (PNoT)*. PNoT leverages process mining techniques to embed business logic within a formal model. By combining the strengths of formal process modeling with advanced prompting techniques, we aim to more effectively capture and replicate real-world decision-making in a controlled, analyzable way.

3 The Petri Net of Thoughts (PNoT) Prompting Paradigm

Our novel paradigm integrates formal process modeling with LLM prompting to create the *Petri Net of Thoughts (PNoT)*. This approach is composed of two major components:

1. **Process Model Construction:** We capture the business logic either by directly incorporating an expert-designed process model or by discovering one from event logs using PM4PY [BvZS23]. The resulting Petri net consists of (A) **places (states)** that represent the distinct states or conditions in the business process, and (B) **transitions** that denote the events or decision points that lead to state changes.
2. **Orchestration of the Prompting Chain:** We orchestrate the prompting chain of the LLM (LLaMA [To23]) by executing the Petri net. Each token replay step corresponds to the progression through a place and the activation of a transition, where the LLM is guided by system prompts that encapsulate the semantics of the corresponding state or event.

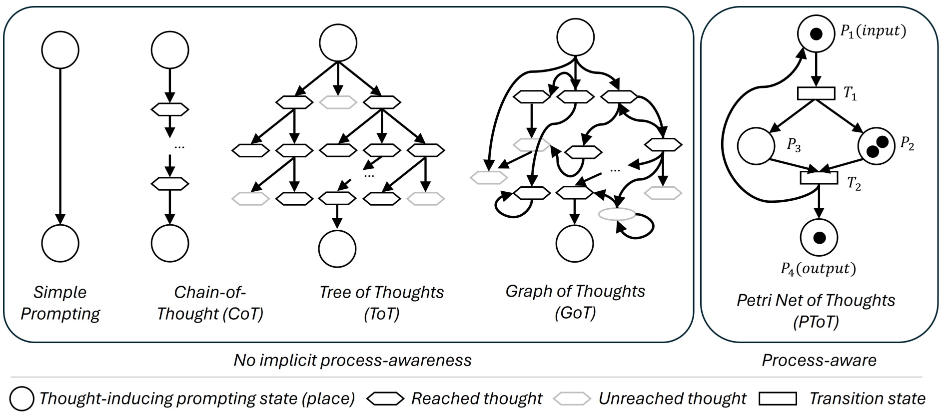

Fig. 1: Conceptual Overview: Orchestrating LLM Prompting with a Petri Net compared to other structure-enhanced prompting approaches

Fig. 1 conceptually illustrates how the Petri net guides the prompting process. The *token replay algorithm* [VDA12] simulates the execution of the Petri net by sequentially activating transitions based on the current marking (i.e., distribution of tokens over places). At each transition, the LLM receives a system prompt that reflects both the state (place) and the decision (transition) to be executed. Below is the pseudocode for prompt orchestration based on the token replay:

Algorithm 1 Petri Net of Thoughts (PNoT)

```
procedure PNoT(InitialMarking, PetriNet, EventLog)
    currentMarking ← InitialMarking
    while not IsFinalMarking(currentMarking, PetriNet) do
        enabledTransitions ← GetEnabledTransitions(currentMarking, PetriNet)
        if enabledTransitions = ∅ then
            break                                   ▷ No transitions available; exit loop
        for all transition in enabledTransitions do
            prompt ← ComposePrompt(currentMarking, transition)
            response ← LLM_Prompt(prompt)
            LogEvent(EventLog, currentMarking, transition, response)
            currentMarking ← FireTransition(currentMarking, transition, PetriNet)
```

The function `ComposePrompt` in the pseudocode constructs a prompt that integrates both the information from the current place (state) and the specifics of the transition (decision), ensuring that the LLM receives a clear, context-rich directive at every step.

At the **place level**, the `ComposePrompt` function aggregates information that captures a summary of the events that have occurred up to the current state. This involves extracting key insights from the event log, synthesizing the rationale behind past decisions, and presenting a concise overview of the context. By summarizing what has happened, the prompt helps the LLM recall and maintain coherence over the reasoning process, ensuring that the model is well-informed about the established business logic and the current state of the process. This summarization acts as an anchor, reinforcing the sequence of events and the contextual background necessary for making informed decisions.

For the **transition level**, the `ComposePrompt` function is designed to articulate the reasoning required to move from the current state to the next business state. This step involves formulating a clear, directive prompt that specifies the decision point at hand, the potential options, and the criteria or constraints drawn from the business process model. Here, the prompt explicitly challenges the LLM to evaluate how to bridge the current state to a desired future state, engaging in a form of guided reasoning that mimics Socratic questioning and thought propagation. The resulting prompt not only instructs the LLM on the necessary analytical steps but also ensures that the transition is executed in alignment with the underlying process logic. Unlike the Graph of Thoughts approach, which allows for flexible and expressive reasoning paths, PNoT uses the formal semantics of Petri nets to encode decision logic, concurrency, and causality explicitly—offering more structured and transparent reasoning aligned with real-world business processes.

Running Example: A Simple Decision Process

Petri Net Structure. Consider a simplified business process for evaluating a new project proposal. The Petri net consists of three key places and two transitions:

- **Places (States): P1: Project Proposal Received** – The initial state where the proposal is first logged. **P2: Proposal Under Review** – The intermediate state where key insights from the proposal are summarized. **P3: Decision Made** – The final state representing the conclusion of the evaluation process.

- **Transitions (Decision Points): T1: Initiate Review** – Moves the process from *Project Proposal Received* to *Proposal Under Review*. **T2: Approve or Reject Proposal** – Transitions from *Proposal Under Review* to *Decision Made*.

Translation to the reasoning process. In the prompt orchestration, a system prompt is responsible for interpreting the initial context and adding a system prompt that corresponds to a place or a transition. This aim to ensure that the LLM appropriately structures its response based on the nature of the state within the process: (1) if it is a place, summarize the accumulated knowledge up to this state; (2) if it is a transition, generate a directive to guide the next step in the reasoning process. Unlike Graph of Thoughts, which allows arbitrary branching and often lacks grounded semantics, PNoT constrains the reasoning trajectory through formally defined transitions and decision logic—supporting more transparent, reproducible, and domain-aligned reasoning paths.

4 Conclusion and Future Work

We introduced *Petri Net of Thoughts (PNoT)*, a novel structured prompting approach that integrates process mining with LLM-based reasoning. PNoT leverages the formalism of Petri nets to enforce structured, concurrent, and verifiable reasoning paths. Future work includes (1) enabling online refinement of the Petri net as the LLM encounters novel reasoning scenarios, (2) integrating domain-specific knowledge into the Petri net to guide tool-augmented reasoning, (3) investigating advanced process mining algorithms for more robust discovery of complex reasoning structures, and (4) extending PNoT for real-world applications where safety and compliance are critical. PNoT represents a promising step towards more systematic and transparent AI reasoning by facilitating existing domain knowledge formalized in conceptual models.

Bibliography

[Be24] Besta, Maciej; Blach, Nils; Kubicek, Ales; Gerstenberger, Robert; Podstawski, Michal; Gianinazzi, Lukas; Gajda, Joanna; Lehmann, Tomasz; Niewiadomski, Hubert; Nyczyk, Piotr; Hoefler, Torsten: Graph of Thoughts: Solving Elaborate Problems with Large Language Models. Proceedings of the AAAI Conference on Artificial Intelligence, 38(16):17682–17690, March 2024.

[BvZS23] Berti, Alessandro; van Zelst, Sebastiaan; Schuster, Daniel: PM4Py: A process mining library for Python. Software Impacts, 17:100556, 2023.

[CBS23] Chaaben, Meriem Ben; Burgueño, Lola; Sahraoui, Houari: Towards using Few-Shot Prompt Learning for Automating Model Completion. In: 2023 IEEE/ACM 45th International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER). pp. 7–12, 2023.

[Di24] Ding, Ruomeng; Zhang, Chaoyun; Wang, Lu; Xu, Yong; Ma, Minghua; Zhang, Wei; Qin, Si; Rajmohan, Saravan; Lin, Qingwei; Zhang, Dongmei: Everything of Thoughts: Defying the Law of Penrose Triangle for Thought Generation, 2024.

[Fi24] Fill, Hans-Georg; Härer, Felix; Vasic, Iva; Borcard, Daniel; Reitemeyer, Benedikt; Muff, Fabian; Curty, Simon; Bühlmann, Marcel: CMAG: A Framework for Conceptual Model Augmented Generative Artificial Intelligence. 12 2024.

[Hu23] Hu, Pengbo; Qi, Ji; Li, Xingyu; Li, Hong; Wang, Xinqi; Quan, Bing; Wang, Ruiyu; Zhou, Yi: Tree-of-Mixed-Thought: Combining Fast and Slow Thinking for Multi-hop Visual Reasoning, 2023.

[Mu89] Murata, T.: Petri nets: Properties, analysis and applications. Proceedings of the IEEE, 77(4):541–580, 1989.

[MX23] Mo, Shentong; Xin, Miao: Tree of Uncertain Thoughts Reasoning for Large Language Models, 2023.

[Ni24] Ning, Xuefei; Lin, Zinan; Zhou, Zixuan; Wang, Zifu; Yang, Huazhong; Wang, Yu: Skeleton-of-Thought: Prompting LLMs for Efficient Parallel Generation, 2024.

[Qi23] Qi, Jingyuan; Xu, Zhiyang; Shen, Ying; Liu, Minqian; Jin, Di; Wang, Qifan; Huang, Lifu: The Art of SOCRATIC QUESTIONING: Recursive Thinking with Large Language Models, 2023.

[Re85] Reisig, Wolfgang: Petri nets-an introduction, volume 4. Springer Verlag, 1985.

[To23] Touvron, Hugo; Martin, Louis; Stone, Kevin; Albert, Peter; Almahairi, Amjad; Babaei, Yasmine; Bashlykov, Nikolay; Batra, Soumya; Bhargava, Prajjwal; Bhosale, Shruti et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288, 2023.

[VDA12] Van Der Aalst, Wil: Process mining, volume 55. ACM New York, NY, USA, 2012.

[vdAWM04] van der Aalst, W.; Weijters, T.; Maruster, L.: Workflow mining: discovering process models from event logs. IEEE Transactions on Knowledge and Data Engineering, 16(9):1128–1142, 2004.

[We23] Wei, Jason; Wang, Xuezhi; Schuurmans, Dale; Bosma, Maarten; Ichter, Brian; Xia, Fei; Chi, Ed; Le, Quoc; Zhou, Denny: Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, 2023.

[Ya23a] Yao, Fanglong; Tian, Changyuan; Liu, Jintao; Zhang, Zequn; Liu, Qing; Jin, Li; Li, Shuchao; Li, Xiaoyu; Sun, Xian: Thinking Like an Expert:Multimodal Hypergraph-of-Thought (HoT) Reasoning to boost Foundation Modals, 2023.

[Ya23b] Yao, Shunyu; Yu, Dian; Zhao, Jeffrey; Shafran, Izhak; Griffiths, Thomas L.; Cao, Yuan; Narasimhan, Karthik: Tree of Thoughts: Deliberate Problem Solving with Large Language Models, 2023.

[YHY24] Yu, Junchi; He, Ran; Ying, Rex: Thought Propagation: An Analogical Approach to Complex Reasoning with Large Language Models, 2024.