UNIVERSITY OF TARTU
INSTITUTE OF COMPUTER SCIENCE
COMPUTER SCIENCE CURRICULUM

Karl Mattias Pärloja

Designing Textual Representations of Business Process Model and Notation for Large Language Models: A Research in Prompting Strategies and Automated Evaluation

Bachelor's thesis (9 ECTS)

Supervisor:
David Chapela De La Campa, PhD

TARTU 2025

Designing Textual Representations of Business Process Model and Notation for Large Language Models: A Research in Prompting Strategies and Automated Evaluation

Abstract:

Business Process Model and Notation (BPMN) is a widely used standard for modelling and analysing business processes. However, its graphical nature and the complexity of its underlying XML representation pose challenges for seamless integration with Large Language Models (LLMs), which primarily process textual data. The objective of this thesis is twofold: to explore the design and development of optimised textual representations by systematically converting BPMN models into structured textual formats to enhance LLM comprehension and to design a benchmark to automatically evaluate the effectiveness of different prompt strategies in enabling LLMs to interpret BPMN structures.

Keywords:

Business Process Model and Notation, Large Language Models, Input Optimisation

CERCS: P175, Informatics, system theory

Äriprotsessimudeli ja -notatsiooni tekstiline esitus suurtele keelemudelitele: uuring sisendistrateegiateks ja automatiseeritud hindamiseks

Lühikokkuvõte:

Äriprotsessimudel ja -notatsioon (BPMN) on laialdaselt kasutusel olev standard äriprotsesside modelleerimiseks ja analüüsimiseks. Siiski tekib suurtel keelemudelitel (LLM-id) äriprotsessimudelitega töötamisel raskuseid, just nende graafilise olemuse ja keeruka XML-põhise struktuuri tõttu. Käesoleval tööl on kaks eesmärki. Esimeseks eesmärgiks on uurida ning disainida optimeeritud tekstilisi esitusi, teisendades BPMN-mudelid süstemaatiliselt struktureeritud tekstivormingutesse, lootuses parandada suurte keelemudelite arusaama antud protsessidest. Teiseks eesmärgiks on luua võrdlusalus, mille abil saab automaatselt hinnata erinevate sisendite tõhusust protsessistruktuuride tõlgendamisel suurte keelemudelite poolt.

Võtmesõnad:
Äriprotsessimudel ja -notatsioon, Suured keelemudelid, Sisendi optimeerimine

CERCS: P175, Informaatika, süsteemiteooria

Table of contents

Introduction

In an era where artificial intelligence continues to redefine the boundaries of automation and decision-making, the integration of Business Process Model and Notation (BPMN) with Large Language Models (LLMs) forms a fertile area of investigation. BPMN, the de facto standard for graphically modelling complex business workflows, is widely used but becomes challenging to work with when combined with text-based AI systems. Simultaneously, the current LLMs exhibit strong capability to understand and reason in natural language, but they are not appropriate for directly interpreting the fine-grained graphical structures.

This thesis starts out exploring the space between these two worlds. I investigate the "design space" of prompt engineering strategies, converting the BPMN diagram to a text-based form more suitable for LLM. In particular, I define and discuss four representations to strike a balance between structural faithfulness and linguistic simplicity. I also systematically develop prompts from BPMN XML to prompt the process for large process models in a reproducible manner.

To provide a fair evaluation of these text representations, I introduce an extendable evaluation benchmark. Built upon a cross-examination approach, my pipeline evaluates the LLM for accuracy and reasoning depth. This benchmark is not only to test my methods, but can also serve as a benchmark in future comparative studies in a similar space.

I experiment with GPT-4o and reveal how different representations impact the model's understanding of processes of varying complexities. My results shed light on practical considerations: while human-readable, narrative prompts are effective in a simple setting, retaining full structural information becomes critical with increasing task complexity. I also look at the interplay between semantic familiarity and formal reasoning by anonymising task labels.

1. Background

This section provides the background of the basic concepts and technologies covered in this thesis. It begins with an introduction to large language models (LLMs), focusing on their abilities, use cases and importance in modern artificial intelligence. The section continues with business process management (BPM), touching on its main principles, lifecycle and the role it plays in organisational efficiency. Together, these paragraphs provide a basis for a better understanding of the collaboration between LLMs and BPM, which is the primary topic of this research.

1.1. Large Language Models (LLMs)

Artificial Intelligence, or AI as it's commonly referred to, is the imitation of human thinking, planning, learning or analysis through a computer or technological solution. This means that a computer or other technological machine behaves like a human and imitates human behaviour - it talks and thinks like one. Artificial intelligence, like humans, is capable of learning by doing the same as we do - learning from the vast amount of data it encounters [1]. The first point of contact for a normal human is conversational robots, which are usually LLMs (large language models) - the AI understands what you are asking and will also give you a textual response in human language.

A Large Language Model (LLM) is a type of artificial intelligence that is designed to understand and produce human language. It is called "large" because it is trained on a very large amount of textual data. We can draw a parallel that, in the same way, we as humans learn to read and write based on the text we see in books, LLM learns to predict the next word in a sentence based on the words that come before it [2]. LLMs already apply across diverse specialities from conversational and pedagogical tools to more sophisticated applications. In the medical field, LLMs could be used, for example, to advance medical learning and facilitate clinical decision-making [3].

Tokenisation and context windows are both important in order to understand how LLMs process and generate language. Tokenisation is a process of breaking down text into smaller chunks (tokens), which can be words, subwords or even just characters. That's important to know because LLMs do not process raw text directly. They interpret and generate sequences of tokens, which are mapped to numerical representations [4]. Tokenisation also influences

how LLMs handle rare words, multilingual inputs and syntactic ambiguity.

Another important concept is the context window, which defines the maximum number of tokens an LLM can consider at once during processing. LLM can “remember” a set amount of tokens of prior text in a conversation or document. This window governs the model’s ability to maintain coherence over long interactions or analyse lengthy texts. However, exceeding this limit forces the model to truncate or “forget” earlier tokens, potentially losing critical context [4].

So LLMs are proficient in producing coherent text, understanding context and translating languages, which makes them useful in fields that require extensive data analysis and synthesis [5]. Different methods can improve LLM performance, including prompt engineering and context augmentation [6]. On the other hand, mitigating bias remains an issue [7]. Addressing these concerns is important given the demand for LLM applications in sensitive domains such as healthcare and legal issues.

1.2. Business Process Management (BPM)

Business Process Management (BPM) is a way to oversee how work is performed in an organisation to ensure consistent outcomes and to take advantage of improvement opportunities. It involves managing entire chains of events, activities and decisions that ultimately add value to the organisation and its customers [8]. For example, in a loan application process, by using BPM, we could help define the structure of the process, monitor its execution and performance and identify bottlenecks. By analysing each step of the loan approval journey (such as application submission, credit checks and decision making), BPM can push us towards smoother operations, reduce processing times and improve customer satisfaction by ensuring that the application is handled efficiently.

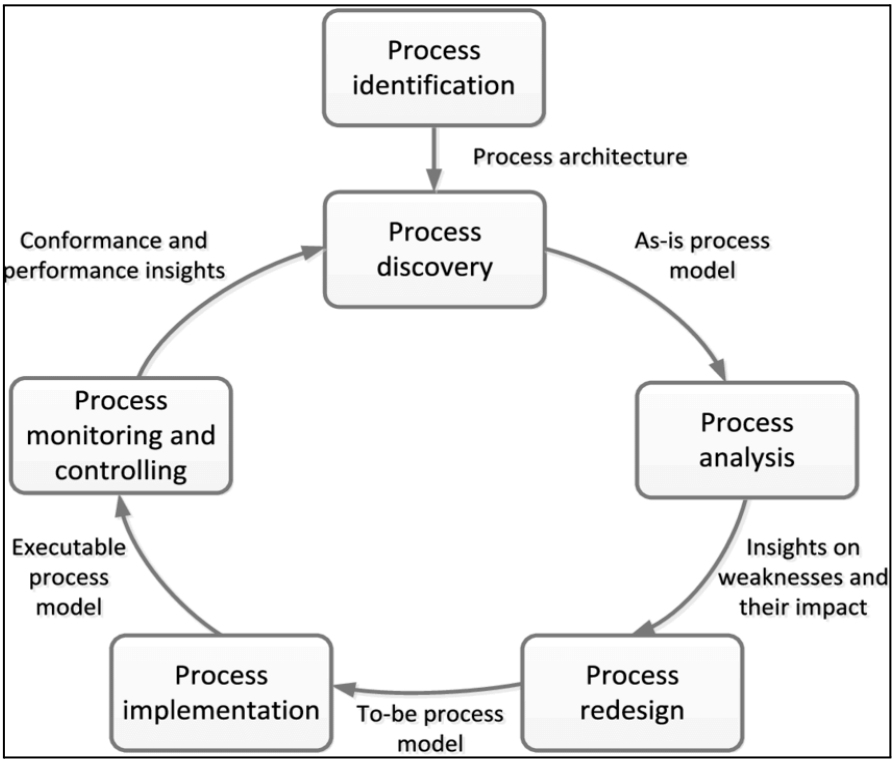

Figure 1: BPM Lifecycle [9]

The BPM lifecycle encompasses a series of phases that help organisations manage and improve their business processes. This type of structured approach helps to ensure that each aspect of the process is effectively addressed, leading to optimal performance. The BPM lifecycle is typically built as stages (Figure 1). Process identification is where relevant processes are recognised and their scope is defined. Then comes process discovery, which involves mapping current processes to understand how they operate. After discovery comes analysis, where performance metrics and potential inefficiencies are evaluated. In the process design phase, improvements and redesigns are formulated. After design comes implementation, entailing the execution of the redesigned processes. Process monitoring is where the ongoing performance is tracked and assessed. Finally comes process refinement, which focuses on continuous improvement by making adjustments based on feedback and changing organisational needs [9].

Most of the stages in the BPM lifecycle work with a process model, which is basically a representation of the activities in the process and the relations among them. A typical standard for this representation is Business Process Model and Notation (BPMN).

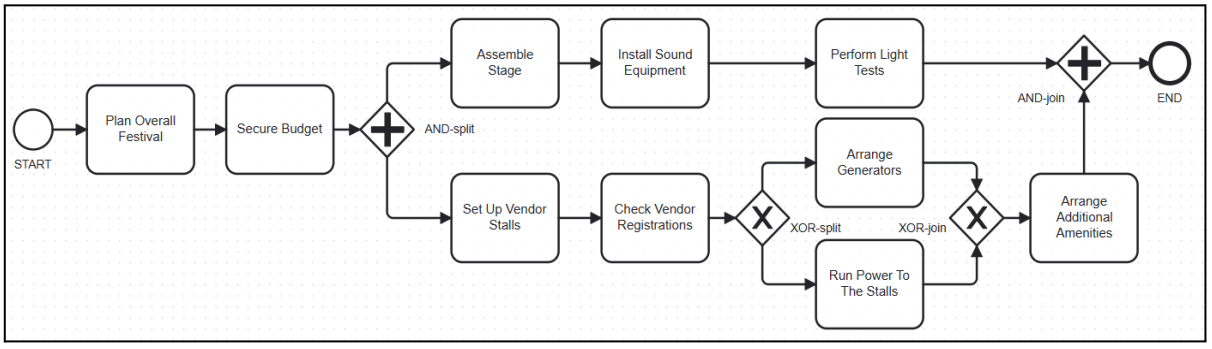

Figure 2: Example BPMN model of Indie music festival preparations

BPMN provides a graphical notation (Figure 2) that is easy to understand for both technical and non-technical stakeholders [10]. It uses standardised symbols to represent various elements of a process, such as tasks, events, gateways and flows, which allows for a clear depiction of the process. BPMN aims to bridge the gap between the design and implementation of business processes by providing a common language for all stakeholders.

The usual BPM stakeholders are process owners, business analysts, IT contributors and upper management. They can all leverage BPMN models for different purposes- from strategic planning to operational execution [10]. This kind of collaboration makes sure that all the process aspects are considered, leading to more informed decision-making and strategic alignment. Therefore, BPM ensures that business processes remain efficient, quality-focused and aligned with market demands [9]. Taking all of the above into account, I have highlighted the pivotal role of BPM in contemporary organisations.

1.3. Large Language Models for Business Process Management

Since the appearance of LLMs, there has been a surge of research exploring their applications across various domains. Their impressive reasoning capabilities and adaptability offer promising avenues for enhancing BPM practices, which often rely on diverse, unstructured textual data. Studies have already investigated the feasibility of conversational process modelling using LLMs [11], demonstrating a shift towards more interactive and adaptive BPM tools.

LLMs could, potentially, enhance Business Process Management (BPM) by addressing various challenges throughout the BPM lifecycle (Figure 1). For example, automate the

identification and analysis of process-related issues by analysing large amounts of unstructured text from customer feedback and internal communications, thereby spotting patterns and summarising complaints [12]. In the redesign and optimisation phases, historical data may help LLMs suggest ways to improve business processes even further [13]. By fine-tuning these models specifically for BPM tasks, organisations are better equipped to tackle the challenges of their operational workflows.

There's already significant promise in the discovery stage of BPM by assisting in the extraction and synthesis of critical business process knowledge from unstructured data sources. For example, recent research highlights the potential of LLMs to streamline the process discovery by analysing event logs, thereby providing insights into process structures and flow dependencies through advanced natural language processing techniques [14]. Moreover, these models can augment traditional process mining techniques, which often focus on time precedence, by incorporating causal relationships between process activities to enhance the identification of intervention points for process improvement [15].

Various techniques have emerged that leverage LLMs to enhance the usability of process mining approaches for stakeholders and managers with limited expertise. For instance, Berti et al. utilised the GPT-4 (OpenAI LLM model) in order to convert natural language inquiries into SQL queries, allowing users to retrieve and understand data without requiring proficiency in technical query languages [16]. Similarly, Jessen et al. applied LLMs to transform natural language questions into executable SQL queries in order to identify bottlenecks in processes [17]. These techniques exemplify the potential of LLMs to make process mining outputs more accessible and actionable for non-expert users, ultimately facilitating better decision-making and process optimisation.

And so, the recent advancements in business process management leverage LLMs in two key ways. Some approaches aim to replace existing algorithms with LLM-driven methods, seeking improved accuracy and efficiency in process analysis. Others focus on enhancing interpretability, assisting users in understanding results when expertise or skill set is lacking. For example, using LLMs to translate complex process metrics into user-friendly insights enables stakeholders without technical expertise to understand performance variations and make informed decisions [18], thus bridging the gap between technical complexity and practical application.

2. Related work

A lot of research has been done to explore prompting strategies, as it plays a critical role in ensuring accurate responses from LLMs, as inadequate prompts can lead to misleading outputs [19]. Prompt engineering has also been studied within the context of Business Process Management and Process Mining (PM). In particular, Berti et al. investigated a range of prompting strategies designed to translate traditional and object-centric process mining artefacts (such as event logs and process models) into textual representations interpretable by large language models. Their findings demonstrate that LLMs have a strong grasp of key process mining abstractions, highlighting their capabilities in handling object-centric perspectives and concepts in process mining [20], suggesting promising directions for us to expand on while contributing to the dialogue surrounding effective interaction with LLMs.

In their work, Berti et al. propose and evaluate three main prompting strategies (direct answering, multi-prompt answering and SQL query generation) in order to interpret and analyse traditional and object-centric process models using LLMs [20]. In this thesis, I focus on process models represented in the BPMN formalism, which is widely used in business process modelling due to its ease of understanding for business stakeholders. I take inspiration from Berti et al. by adapting their proposed methods as I introduce a set of novel prompting techniques aimed at better capturing the semantics and structure of BPMN models, expanding the applicability of LLMs in business-oriented process analysis.

Regarding the second contribution of this thesis (the evaluation of the prompting strategies), previous work on the use of LLMs in process mining has mostly followed two main approaches. Some studies, such as Berti's, rely on manual evaluation, where experts assess the quality and correctness of the LLM responses based on a limited set of queries and scenarios [20]. However, more recent works [21][22][23] introduce automated evaluation methods that follow two main directions:

1. Defining a set of questions along with deterministic ground truth answers and then automatically checking whether the LLM outputs match these expected answers.

2. Collecting expert-validated answers and then using an independent LLM instance to

compare and assess the correctness of a given LLM response against the reference.

For this thesis, I adopt the first approach by designing a set of questions and corresponding expected answers to carry out a direct evaluation of the prompting strategies.

In summary, I focus on creating descriptive prompts for BPMN and validating their effectiveness by cross-examining large language models. Building on previous research, I try to advance methods that improve clarity and accuracy in LLM responses. My thesis adopts BPMN because it is widely familiar and practical in a business context. The focus is to develop a more comprehensive, automated benchmark that can be readily adapted to various LLMs, ultimately broadening the applicability of prompt engineering practices for process models.

3. Methodology

My practical experience working with BPMN served as the initial motivation for this thesis. In particular, I frequently encountered difficulties when attempting to leverage LLMs for comprehending BPMN structures, which gave me reason for further research into this topic.

Early in the project, I began working under the supervision of David, whose extensive expertise in Business Process Management (BPM) and LLMs proved invaluable. Over the first two months, my efforts focused on broad foundational research, starting with core concepts in BPM and process mining, then exploring LLMs and prompt engineering techniques. David's guidance helped me identify both academic and practical resources, laying the groundwork for much of the literature referenced in this thesis. Once I had established a solid understanding of these areas, I examined existing studies on the integration of process modelling and LLMs. Ultimately, I chose to narrow my scope to BPMN, given its wide adoption in business management and its relevance to real-world applications.

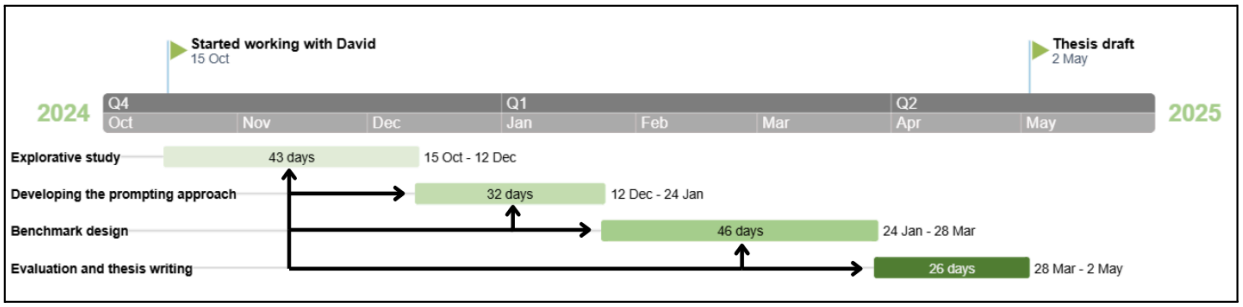

Figure 3: Thesis timeline

As shown in Figure 3, the project progressed through four iterative phases. Initially, an exploratory study was conducted. It involved a thorough examination of literature related to BPM, process mining and LLM. This review addressed unresolved questions and defined the project's scope. Next, I focused on crafting the prompts. Here, the Python functions were developed that converted raw BPMN XML into natural-language prompts, refining them after pilot tests. Following this, I designed the evaluation benchmark. An automated evaluation pipeline was developed to assess the LLMs understanding of models. The final phase involved executing the evaluation pipeline, analysing the results and consolidating

insights into the completed thesis.

To prevent becoming stuck at any one stage, the project intentionally adopted an iterative workflow. Each time implementation or evaluation raised new inquiries, I revisited existing literature and adjusted prior assumptions before progressing, thereby ensuring that each phase was guided by the most precise and thorough understanding attainable.

3.1. Research Approach and Workflow

The core research objective was to evaluate the ability of LLMs to understand BPMN-based prompts. To structure this investigation, I defined four distinct prompting strategies. Additionally, I designed an evaluation pipeline in order to assess how well the LLMs comprehended the underlying process structure by comparing their responses to predefined ground truths. In the following two months, I developed a set of helper functions to convert .bpmn XML format into a more human-readable, natural language format suitable for prompting LLMs. To develop prompts from BPMN diagrams, I first converted each BPMN file into a directed graph that encapsulated the flow of tasks, events and gateways. Building on this graph, I used helper functions to translate its nodes and edges into clear, structured statements. By systematically describing each process element (such as tasks and gateways), I can accurately reflect the BPMN logic, which can be used directly by LLMs. This laid the groundwork for the experimentation phase.

For experimentation, I curated a diverse collection of BPMN models, which I categorised into three complexity levels based on structural features and size. To assess the LLM's understanding across these models, I designed five groups of evaluation questions inspired by the LTL (Linear Temporal Logic). These question groups were structured to target specific aspects of comprehension.

With these resources in place, I implemented a validation pipeline using LangChain. This pipeline automated the evaluation process by taking a BPMN model, generating prompts according to the four strategies, posing the corresponding questions with expected answers and recording the LLM responses for analysis. This development phase lasted approximately three months.

Throughout the academic year, I met with David every Wednesday to discuss progress, challenges and next steps. The overall workflow was iterative. I would revisit and refine the earlier stages when edge cases were discovered during prompt generation or model testing. Similarly, as new models were introduced, I expanded the set of evaluation questions to accommodate emerging patterns and cases. Rather than following a strictly linear methodology, each step informed and enriched the others in a continuous development loop.

4. Prompting Approach

The four prompt types used in this research are raw BPMN, simplified BPMN, Humanified BPMN and directly-follows graph (DFG). Prompt types were designed and selected through discussions with my supervisor and informed by observations from the literature. Each represents a different way of encoding or abstracting a BPMN model to uncover how LLMs respond to syntactic versus semantic representations of processes.

The process begins with BPMN files in their original .bpmn XML format. These files contain structured elements such as tasks, gateways and events, each defined by their type and assigned an identifier. Using these notations, I parse the XML to extract all relevant elements and build a semantic representation of the process. To improve human readability, I extract and display the name attributes associated with tasks rather than their IDs. This helps ensure that the generated prompts are semantically interpretable, especially in the Humanified and simplified prompt formats. All elements are used to construct a directed graph representation of the process. Although each of the four prompting strategies has its own output style, they all rely on the same underlying graph. The formatting into prompt texts is handled by dedicated prompt generation functions, each tailored to a specific prompt style. While these functions automate the majority of the generation process, manual verification is performed for each prompt.

All generated prompts are saved into a dedicated prompts folder within the corresponding process directory. Each model is stored under its own named folder in a structured file system to be correctly handled by the evaluation pipeline.

4.1. Raw BPMN

Raw BPMN represents the unmodified XML structure, preserving the complete XML structure that describes tasks, events, gateways and sequence flows. This format is highly structured and detailed, but difficult for an LLM to interpret without a specialised understanding of BPMN's XML schema. Because early tests ran into context-window limits, I also simplified the XML itself. I did apply minimal cleanup to remove visual metadata and other non-structural elements, retaining only the contents of the <process> tag. My aim was to retain the critical information that defines the process flow while avoiding unnecessarily

long prompts, which can be both expensive and time-consuming to process at scale. In this approach, the LLM is still required to handle a complex, machine-readable format, simulating a real-world automation scenario where a BPMN model might be directly passed to an LLM.

```
<process id="ExclusiveGatewayThreeBranches" isExecutable="true">
    <startEvent id="startEvent" name="Start" />
    <userTask id="taskA" name="Check Form" />
    <exclusiveGateway id="exclusiveGateway" name="Determine Priority" />
    <userTask id="taskB" name="Low Priority Handling" />
    <endEvent id="endEvent" name="End" />
    <!-- Sequence Flows -->
    <sequenceFlow id="flow1" sourceRef="startEvent" targetRef="taskA" />
    <sequenceFlow id="flow2" sourceRef="taskA" targetRef="exclusiveGateway" />
    <sequenceFlow id="flow3" sourceRef="exclusiveGateway" targetRef="taskB">
    <conditionExpression  xsi:type="tFormalExpression">{condition_for_B}</conditionExpression>
    </sequenceFlow>
    <sequenceFlow id="flow4" sourceRef="taskB" targetRef="endEvent" />
  </process>
```

Figure 4: Raw BPMN snippet

Figure 4 illustrates an example of the raw BPMN XML and the structural details that remain after cleanup.

4.2. Simplified BPMN

The design of Simplified BPMN is based on the textual-abstraction method proposed by Alessandro Berti and his team in their research on PetriNets [16]. Simplified BPMN provides a textual breakdown of the BPMN elements (tasks, events, gateways and flows) using domain-specific explanations. The goal here is to maintain structural fidelity while translating the raw XML into more readable, categorised descriptions. This prompt is generated by a Python function, which operates by traversing the underlying directed graph structure and extracting its nodes and edges into coherent textual sections. The code begins by appending an introductory line to frame the content. Next, it iterates through all graph nodes, filtering them by their type attribute and then appending each relevant node's label alongside a brief descriptor to the output. For instance, tasks are listed under "The tasks in a BPMN model…" to signal that each item is a distinct activity. Events are similarly grouped and gateways include short definitions for Parallel (AND) and Exclusive (OR) logic. Once the node information is gathered, the code moves on to the edges, which are iterated and formatted in a simple "Source -> Target" notation.

```
The tasks in a BPMN model represent specific activities within the business process. Each task
signifies a step that must be performed to complete the process. The tasks included in this BPMN
model are:

Client segmentation (task)
...

Events in a BPMN model represent occurrences that affect the flow of the process. They can signify
the start, intermediate, or end of a process. In this BPMN model, the events are:

StartEvent_1pnvxlx (START)
Event_0o1yxen (END)

Gateways in a BPMN model control the process flow by either diverging into multiple paths or
converging different paths back together. There are two types of gateways:
- **Parallel Gateway (AND):** This type can only be fired when all its incoming flows are 'active',
and its firing activates all its outgoing flows.
- **Exclusive Gateway (OR):** This type allows only one of its outgoing flows to be taken based on
predefined conditions. The choice of which flow to take is determined at runtime.

Gateway_1l2ddi4 (AND-split)
Gateway_00im1mk (OR-split)

Flows in a BPMN model define the sequence of execution for tasks, events, and gateways. The flows in
this model are:

START -> Client segmentation
Client segmentation -> AND-split (Gateway_1l2ddi4)
...
Prepare public communication -> END (Event_0o1yxen)
```

Figure 5: Pseudo-figure illustrating the text-based structure of a Simplified BPMN prompt.

Figure 5 illustrates that the output includes separate lists for tasks, events, gateways and the flows connecting them. By structuring the prompt in this way, balancing between technical detail and readability, I hope to find a middle ground for evaluating the LLM's ability to map structured input to understandable process flows.

4.3. Humanified BPMN

In Humanified BPMN, the process is expressed in a natural language format, resembling how a human might describe a workflow. The prompt generation function relies on a breadth-first search (BFS) traversal of the process graph to produce statements like "From [A] to [B]". Join gateways (where multiple incoming flows converge) are removed to reduce complexity and split gateways (parallel or exclusive branching points) are detected and recorded in a mapping structure. Any node recorded as a split is printed as a consolidated statement ("From [A] to [B] AND [C]"), mimicking natural language for concurrent flows. Standard edges are just listed in the order they are discovered. Zero in-degree becomes the starting point and as each node is visited, it's inserted into the output lines.

```
This BPMN process flow represents a structured sequence of tasks. Each task leads to one or more subsequent tasks.

From [START] to [Compose html]
From [Compose html] to [Send to affected client] OR [Send to everyone]
From [Send to affected client] to [END]
From [Send to everyone] to [END]

Explanations for edge types:

"From [A] to [B]" denotes a sequential flow indicating that B is executed after A happened.
"From [A] to [B] and [C]" denotes a flow indicating that both B and C are executed after A happened.
"From [A] to [B] or [C]" denotes a flow indicating that either B or C (but only one) can be executed after A
happened.
```

Figure 6: Example of a Simplified BPMN prompt

As seen in Figure 6, the prompt also includes an introductory line at the beginning and an outro line at the end, giving the LLM additional context. Aim of this format is to see how well an LLM can interpret BPMN models when presented in a human-readable, semantically rich description. It abstracts away formal BPMN constructs while preserving the logical relationships between steps. This style is relevant for business users or non-technical stakeholders who interact with process models in descriptive language rather than formal diagrams.

4.4. DFG

DFG representation consists of two parts: a list of nodes (tasks) and a list of edges indicating direct execution order between them. It is a simplified graph abstraction widely used in process mining to represent process behaviour without explicitly modelling gateways and events. The code to generate this format is also straightforward: I take the directed graph built from the original BPMN, then collect all pairs of nodes that share a direct edge, ignoring gateways altogether.

```
Nodes: Client segmentation, Send to own employee, Send to everyone
Edges: (START, Client segmentation), (Client segmentation,  Send to own employee),
(Client segmentation, Send to everyone), (Send to own employee, END), (Send to
everyone, END)
```

Figure 7: Example of a DFG prompt

This format (Figure 7) is pretty structurally reduced, presenting only the executable edges between tasks. It tests whether LLMs can infer process logic from minimalistic representations and is useful for comparing performance on less context-rich inputs.

5. Benchmark

I use a question-based assessment method to evaluate how well the model understands BPMN structures based on my prompts. First, I organise the BPMN models into three difficulty levels.

- Level 1 includes simple, linear flows with basic gateways, possible concurrency and around 8 activities.
- Level 2 introduces nested structures, longer loops and more complex branching.
- Level 3 consists of highly complex models with many activities, parallel paths, loops and multiple possible end points. Each model in this category contains more than 15 distinct activities, reflecting their substantially greater structural and behavioural complexity.

These levels help us test how the model performs under increasing structural complexity.

For each BPMN model, I generate a curated set of cross-examining questions. These questions are divided into five main groups, each targeting a specific aspect of process understanding: sequential ordering, task co-occurrence and exclusivity, frequency and repetition, causality and dependency and subprocess or case existence. Each model's question set includes a balanced mix from each group to ensure consistent and fair evaluation across different prompt types. Some of the questions are inspired by ideas from linear temporal logic (LTL), as they often ask about event ordering, conditions and logical relationships.

Each question in my evaluation has a predefined ground truth answer (YES or NO), based on a manual analysis of the BPMN model. I ask the LLM to answer each question using only "YES" or "NO", which allows us to directly compare the model’s response to the ground truth. This comparison forms the basis of my accuracy calculation. If the answer matches the expected result, it's considered correct.

To make the final analysis easier, I organise the accuracy scores into a grid. For each prompting type, columns represent the three levels of BPMN model complexity and the rows represent the five different types of questions. This layout creates a kind of map that helps us

visually track which types of understanding the model handles well and where it struggles. For example, I can quickly see if the model is good at answering ordering questions on simple models but fails with causality questions on more complex ones. This makes my evaluation both detailed and accessible.

6. Evaluation

This section evaluates each prompting method described in “Prompting Approach” over the structured benchmark presented in “Benchmark”. The objective is to measure the ability of LLMs in interpreting BPMN processes in the four prompt formats (Raw BPMN, Simplified BPMN, Humanified BPMN, DFG) of different degrees of complexity and question types. I quantitatively measure accuracy by contrasting LLM responses with predetermined 'ground-truth' answers to investigate strengths, limitations and trends of the model performance. The results are presented in a way which allows us to investigate in detail the influence of different prompt styles on comprehension along both the structural and semantic axes. This systematic comparison is designed to uncover best practices for prompting process models for LLM interactions.

6.1. Evaluation Set-up

To apply the proposed evaluation method, I developed an evaluation pipeline designed to cross-examine the LLM (Figure 8).

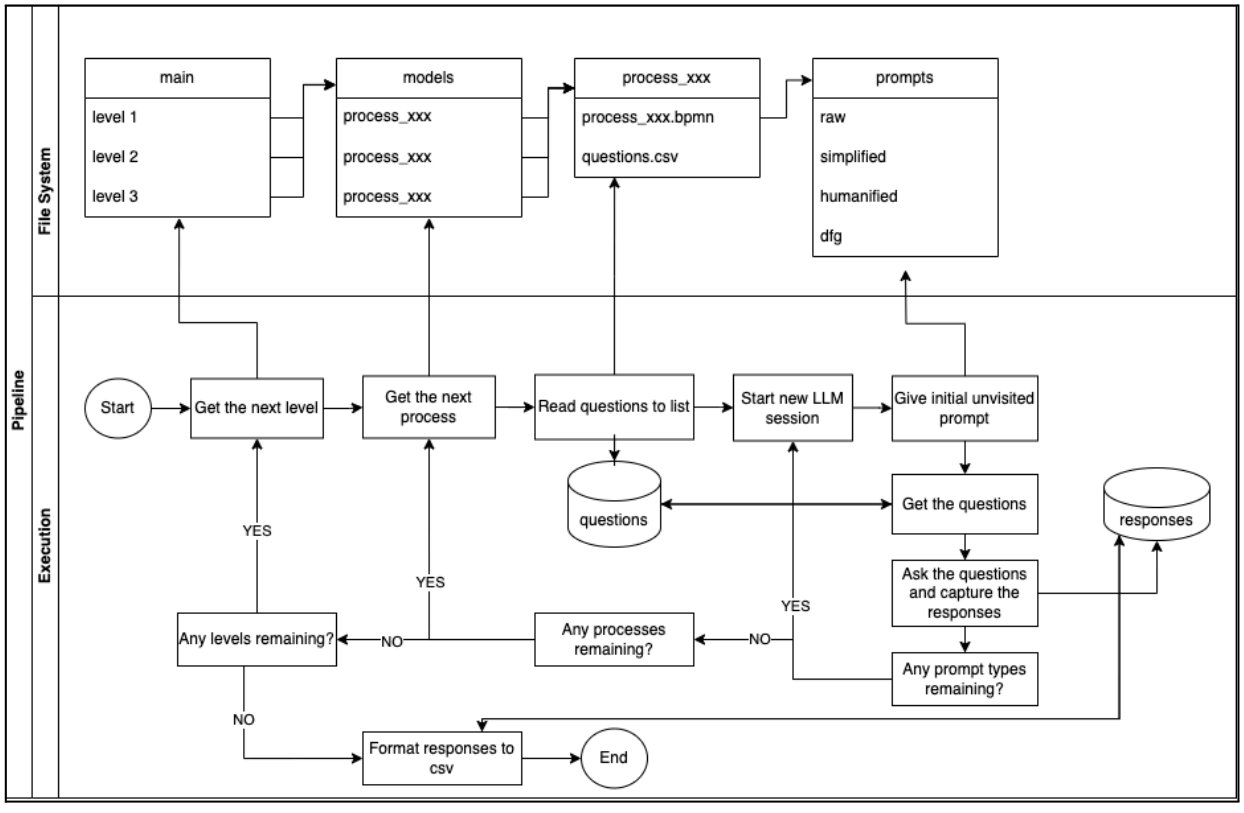

Figure 8: Design of the evaluation pipeline

This pipeline is built using the LangChain[1] framework, which provides a convenient way to interact with language models. By abstracting the LLM interface into standardised components, LangChain facilitates a modular connection. This allows for the easy substitution of different LLM endpoints with minimal configuration adjustments. For this thesis, I selected OpenAI's GPT-4o model due to its availability, reliability and resource efficiency.

Each process folder in the system includes a .bpmn file representing the original process, a prompts subfolder containing the formatted prompts and a questions.csv file containing structured evaluation questions. The questions file has three components for each question: the question category (denoting which aspect of understanding is being tested), the question itself and a ground truth (correct expected answer, either YES or NO).

The pipeline iterates through each process dictionary, parsing the generated prompts and corresponding questions. Then it communicates with an LLM via the LangChain interface to query it using the prompts and questions while capturing its responses.

After cross-examination, results are saved back into the corresponding process folder (to keep the transparency of the results). For each process, a results CSV file is generated that includes the prompt type used, the questions that I asked, LLM's answers and a boolean value indicating whether the response matched the ground truth. In the end, all results are aggregated into a combined master file, which includes the responses for all the models and prompt types. This file is designed to import results into a spreadsheet, where the data can be easily mapped to the previously mentioned grid view.

As a measure of goodness, I report accuracy, calculated as the percentage of evaluation questions for which the model's response exactly matches the ground-truth answer. This simple metric effectively reflects the LLM's ability to generate accurate outputs across the entire test suite and provides a clear foundation for comparing different prompt strategies and model variations.

This modular and file-based structure allows for scalable experimentation. New BPMN

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

models can be added by simply creating a new folder with the appropriate structure, making the system adaptable for future extensions beyond the scope of this thesis. All Python scripts, the related directory structure and a selection of BPMN models created for this thesis are included in the appendix.

6.2. Results

The following tables summarise the accuracy achieved for each combination of question group, prompt type and model complexity.

PROMPTING METHODOLOGY	RAW BPMN	SIMPLIFIED	HUMANIFIED	DFG
MODEL LEVEL	L1	L1	L1	L1
QUESTION				
Control Flow & Ordering	62.50%	50.00%	70.83%	70.83%
Task Co-occurrence & Exclusivity	37.50%	50.00%	50.00%	50.00%
Frequency & Repetition	87.50%	83.33%	87.50%	87.50%
Causality & Dependency	54.17%	58.33%	66.67%	70.83%
Logical Constraints	43.75%	50.00%	56.25%	50.00%
AVARAGE	57.08%	58.33%	66.25%	65.83%

Table 1: Accuracy results by question category and prompting type for level 1 models

Across all five question groups, Level 1 (L1) scores are mostly similar for the strategies (Table 1). The accuracy scores range between 37.50% and 87.50%, with Humanified and DFG prompts often performing even slightly better than Raw and Simplified. For example, for Control Flow & Ordering, Humanified and DFG both reach 70.83%, while Raw stands at 62.50% and Simplified at 50.00%. In Frequency & Repetition, all four prompts perform quite similarly, with accuracies ranging between 83.33% and 87.50%.

PROMPTING METHODOLOGY	RAW BPMN	SIMPLIFIED	HUMANIFIED	DFG
MODEL LEVEL	L2	L2	L2	L2
QUESTION				
Control Flow & Ordering	56.00%	48.00%	56.00%	36.00%
Task Co-occurrence & Exclusivity	60.87%	73.91%	52.17%	82.61%
Frequency & Repetition	41.67%	37.50%	37.50%	41.67%
Causality & Dependency	50.00%	58.33%	54.17%	58.33%
Logical Constraints	40.00%	46.67%	53.33%	46.67%
AVARAGE	49.71%	52.88%	50.63%	53.06%

Table 2: Accuracy results by question category and prompting type for level 2 models

Moving to Level 2 (L2), Table 2 shows a noticeable drop in several categories, yet the relative spread between prompt types remains moderate. For Task Co-occurrence and Exclusivity, Simplified and DFG exceed with 73.91% and 82.61%, respectively, with Raw's accuracy being 60.87% and Humanified's 52.17%. A similar pattern also appears in Causality and Dependency, where the prompt accuracies range between 50.00% and 58.33%.

PROMPTING METHODOLOGY	RAW BPMN	SIMPLIFIED	HUMANIFIED	DFG
MODEL LEVEL	L3	L3	L3	L3
QUESTION				
Control Flow & Ordering	66.67%	54.17%	58.33%	58.33%
Task Co-occurrence & Exclusivity	66.67%	54.17%	45.83%	41.67%
Frequency & Repetition	45.83%	41.67%	37.50%	37.50%
Causality & Dependency	50.00%	29.17%	54.17%	45.83%
Logical Constraints	37.50%	37.50%	43.75%	31.25%
AVARAGE	53.33%	43.33%	47.92%	42.92%

Table 3: Accuracy results by question category and prompting type for level 3 models

The results vary significantly more for Level 3 (L3). Looking at Table 3, we can see that Raw BPMN attains the highest accuracy in Control Flow & ordering, at 66.67%. Simplified, Humanified and DFG accuracies measure 54.17%, 58.33% and 58.33%, respectively. In Task Co-occurrence and exclusivity, Raw again leads at 66.67 %, with the other three prompts ranging from 41.67% to 54.17%. By contrast, the Logical Constraints scores for L3 are low across the board, ranging between 31.25% and 43.75%.

PROMPTING METHODOLOGY	RAW	SIMPLIFIED	HUMANIFIED	DFG
MODEL LEVEL	L3	L3	L3	L3
QUESTION				
Control Flow & Ordering	54.17%	41.67%	45.83%	54.17%
Task Co-occurrence & Exclusivity	50.00%	45.83%	41.67%	41.67%
Frequency & Repetition	54.17%	50.00%	37.50%	37.50%
Causality & Dependency	54.17%	50.00%	58.33%	58.33%
Logical Constraints	50.00%	43.75%	43.75%	43.75%
AVARAGE	52.50%	46.25%	45.42%	47.08%

Table 4: Accuracy results for anonymised testing

Table 4 shows the anonymised Level 3 results, where task labels were replaced with neutral

identifiers instead of having real-life event names. Here, most values decrease relative to the non-anonymised L3 runs. For instance, for Control Flow & Ordering, Raw falls from 66.67% to 54.17%, Simplified from 54.17% to 41.67% and Humanified from 58.33% to 45.83%. A similar shift is visible in Frequency & Repetition, where Raw moves from 45.83% to 54.17% and Simplified and DFG settle at 50.00% and 37.50%, respectively. Humanified stays unchanged.

6.3. Discussion

The first thing that stands out is the steady fall-off in accuracy as we move from Level 1 to Level 3 models. A straightforward explanation would be that each extra gateway, loop or parallel branch increases the number of possible execution paths the LLM has to keep track of. Longer prompts also consume more of the context window, so the LLM has less "mental room" to reason about every path in detail. On top of that, nesting and loops create dependencies that are harder to trace with pattern-matching alone. The model has to simulate the process more explicitly and any small misunderstanding early on can lead to wrong answers later. Put simply, the harder the diagram, the more places there are for GPT-4o to slip up, so the overall accuracy naturally goes down.

Another clear trend in the results for the easier models (Levels 1 and 2) is that the more processed prompts, like full sentences in Humanified and DFG edge lists, consistently beat raw and less processed prompt versions. My interpretation is that, when a process contains only a few branches and almost no nesting, all extra information (like tags or IDs) add noise instead of value. A Humanified approach gives GPT-4o exactly enough information it needs in a single, compact and familiar sentence. Similarly, DFG's edge list style boils the relationship down to only a few tokens that fit neatly in the model's working memory. Because the structure is simple, the LLM can infer ordering, exclusivity and repetition directly from these cues without spending attention and time on parsing IDs, gateways or other metadata. For easier structures, less seems to be more. The more the prompt looks like ordinary text(or the more compact it is), the better the model's accuracy.

However, moving to models with greater structural complexity (Level 3), we see things differently. Here, the full Raw BPMN (cleaned-up XML with every task, gateway and sequenceFlow) comes out on top. My best take is that when a process splits into multiple branches, loops and merge points, reduced prompt formats start dropping critical knowledge.

A Humanified sentence like “From Task A to Task B and Task C or Task D” hides when and how the branches would rejoin and a simple DFG edge says nothing about an intervening gateway or a loop back into the flow. Yet Raw spells out each gateway type and the exact targets of every outgoing flow, giving LLM a bunch of raw material to reconstruct the entire execution logic. Although this type of XML input is very token-heavy, the extra detail appears to outweigh the cost. The model can rely on the rigid tag structure to keep track of complex dependencies instead of guessing from a narrative summary. So, once the structures get messy, having all the structural information (even if in machine-readable form) goes a long way in helping LLM better understand the process and flow logic.

When tasks were anonymised, there appeared to be two key differences. For categories that rely on recognising what activities are done together, accuracy generally fell when compared to non-anonymised results. This suggests that meaningful task names like “Approve Invoice” or “Ship Goods” give LLM common-sense hints about which steps would naturally follow each other or exclude each other. These types of hints disappear when everything is renamed to something ambiguous and not very telling. In contrast, scores for Dependency and Logical Constraints often rise after anonymisation. I find that shorter, neutral labels may actually help. With fewer tokens and no distracting business terms, the model spends less of its context window on wording and more on the gateway structure itself. Overall, removing descriptive names seems to take away from some real-world intuition about ordering and co-occurrence, but it can make purely structural reasoning (especially causal links) quite a bit easier for the LLM.

7. Conclusion

The objective of this thesis was to better understand and promote space between graphical BPMN and text-centric LLMs, which I aimed to achieve by creating, formalising and analysing four different text-representations: Raw BPMN, Simplified BPMN, Humanified BPMN and DFG. By converting BPMN XML to semantic prompts and proposing and applying a benchmark based on cross-examination, I showed how each representation affects LLM understanding and reasoning over 3 tiers of process complexity.

The evaluation using GPT-4o indicated that for simple, linear processes, lean and narrative prompts such as Humanified BPMN and the compact DFG format resulted in the highest accuracy, underlining the role of natural-language fluency and minimal structural noise. With increasing process complexity (nesting gateways, loops, and parallelism), the Raw BPMN format, despite its verbosity, provided the detailed structural cues necessary for correct reasoning, outperforming more abstracted representations. Anonymising task labels also highlights the trade-off between semantic familiarity and pure structural reasoning: task semantics help with identifying co-occurrence and ordering, but can distract the model's ourattention from formal dependencies.

Together, the results provide practical guidance for practitioners seeking to integrate LLMs with BPM tools: short human-oriented prompts for simple workflows and maintain full structural information for complex models. Furthermore, by providing an extensible, file-based evaluation pipeline and benchmark, the present work paves the way for future research on prompt engineering in process mining, including empirical studies across LLM architectures and domain settings. This thesis, in the end, provides both practical understanding and reusable artefacts for the continuous pursuit of easy and accurate LLM-driven process analysis.

References

[1] Jackson PC. Introduction to artificial intelligence. Courier Dover Publications; 2019 Aug 14.

[2] McCoy RT, Smolensky P, Linzen T, Gao J, Celikyilmaz A. How much do language models copy from their training data? evaluating linguistic novelty in text generation using raven. Transactions of the Association for Computational Linguistics. 2023 Jun 29;11:652-70.

[3] Clusmann J, Kolbinger FR, Muti HS, Carrero ZI, Eckardt JN, Laleh NG, Löffler CM, Schwarzkopf SC, Unger M, Veldhuizen GP, Wagner SJ. The future landscape of large language models in medicine. Communications medicine. 2023 Oct 10;3(1):141.

[4] Johnsen M. Large language models (LLMs). Maria Johnsen; 2024 Jun 15.

[5] Hadi MU, Qureshi R, Shah A, Irfan M, Zafar A, Shaikh MB, Akhtar N, Wu J, Mirjalili S. Large language models: a comprehensive survey of its applications, challenges, limitations, and future prospects. Authorea Preprints. 2023 Nov;1:1-26.

[6] Marvin G, Hellen N, Jjingo D, Nakatumba-Nabende J. Prompt engineering in large language models. InInternational conference on data intelligence and cognitive informatics 2023 Jun 27 (pp. 387-402). Singapore: Springer Nature Singapore.

[7] Dai S, Xu C, Xu S, Pang L, Dong Z, Xu J. Bias and unfairness in information retrieval systems: New challenges in the llm era. InProceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2024 Aug 25 (pp. 6437-6447).

[8] Hung RY. Business process management as competitive advantage: a review and empirical study. Total quality management & business excellence. 2006 Jan 1;17(1):21-40.

[9] Dumas M, La Rosa M, Mendling J, Reijers HA. Fundamentals of business process management. Heidelberg: Springer; 2013 Jan.

[10] Chinosi M, Trombetta A. BPMN: An introduction to the standard. Computer Standards & Interfaces. 2012 Jan 1;34(1):124-34.

[11] Köpke J, Safan A. Efficient LLM-based conversational process modeling. InInternational Conference on Business Process Management 2024 Sep 1 (pp. 259-270). Cham: Springer Nature Switzerland.

[12] Vidgof M, Bachhofner S, Mendling J. Large language models for business process management: Opportunities and challenges. InInternational Conference on Business Process Management 2023 Sep 1 (pp. 107-123). Cham: Springer Nature Switzerland.

[13] Mustansir A, Shahzad K, Malik MK. Towards automatic business process redesign: an

NLP based approach to extract redesign suggestions. Automated software engineering. 2022 May;29(1):12.

[14] Žemgulienė J, Valukonis M. Structured literature review on business process performance analysis and evaluation. Entrepreneurship and Sustainability Issues. 2018 Sep 30;6(1):226-52.

[15] Grohs M, Abb L, Elsayed N, Rehse JR. Large language models can accomplish business process management tasks. InInternational Conference on Business Process Management 2023 Sep 11 (pp. 453-465). Cham: Springer Nature Switzerland.

[16] Berti A, Schuster D, van der Aalst WM. Abstractions, scenarios, and prompt definitions for process mining with llms: A case study. InInternational Conference on Business Process Management 2023 Sep 11 (pp. 427-439). Cham: Springer Nature Switzerland.

[17] Jessen U, Sroka M, Fahland D. Chit-chat or deep talk: prompt engineering for process mining. arXiv preprint arXiv:2307.09909. 2023 Jul 19.

[18] Chapela-Campa D, Dumas M. From process mining to augmented process execution. Software and Systems Modeling. 2023 Dec;22(6):1977-86.

[19] Schramm S, Preis S, Metz MC, Jung K, Schmitz-Koep B, Zimmer C, Wiestler B, Hedderich DM, Kim SH. Impact of Multimodal Prompt Elements on Diagnostic Performance of GPT-4V in Challenging Brain MRI Cases. Radiology. 2025 Jan 21;314(1):e240689.

[20] Berti A, Qafari MS. Leveraging large language models (llms) for process mining (technical report). arXiv preprint arXiv:2307.12701. 2023 Jul 24.

[21] Lashkevich K, Milani F, Avramenko M, Dumas M. Llm-assisted optimization of waiting time in business processes: A prompting method. InInternational Conference on Business Process Management 2024 Sep 1 (pp. 474-492). Cham: Springer Nature Switzerland.

[22] Kubrak K, Botchorishvili L, Milani F, Nolte A, Dumas M. Explanatory Capabilities of Large Language Models in Prescriptive Process Monitoring. InInternational Conference on Business Process Management 2024 Sep 1 (pp. 403-420). Cham: Springer Nature Switzerland.

[23] Berti A, Kourani H, van der Aalst WM. PM-LLM-Benchmark: Evaluating large language models on process mining tasks. InInternational Conference on Process Mining 2024 Oct 14 (pp. 610-623). Cham: Springer Nature Switzerland.

Appendices

The codebase, including the corresponding file structure, can be found in the following GitHub repository:

https://github.com/AutomatedProcessImprovement/bpmn-prompt-strategies

Non-exclusive licence to reproduce the thesis and make the thesis public

I, Karl Mattias Pärloja

1. grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the digital archives of the University of Tartu until the expiry of the term of copyright, my thesis **"Designing Accurate Textual Representations of Business Process Model and Notation for Large Language Models: A Research in Prompting Strategies and Automated Evaluation"** supervised by **David Chapela De La Campa**;

2. grant the University of Tartu a permit to make the thesis specified in point 1 available to the public via the web environment of the University of Tartu, including via the digital archives, under the Creative Commons licence CC BY NC ND 4.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright;

3. am aware of the fact that the author retains the rights specified in points 1 and 2;

4. confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Karl Mattias Pärloja
14/05/2025