Decision Support Systems 200 (2026) 114563

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

Leveraging large language models for enhanced process model comprehension☆

Humam Kourani [a,b,*], Alessandro Berti [a,b], Jasmin Hennrich [a], Wolfgang Kratsch [a], Robin Weidlich [a], Chiao-Yun Li [a,b], Ahmad Arslan [a,b], Wil M.P. van der Aalst [a,b], Daniel Schuster [a,b]

[a] Fraunhofer Institute for Applied Information Technology FIT, Schloss Birlinghoven, Sankt Augustin, 53757, Germany
[b] RWTH Aachen University, Ahornstraße 55, Aachen, 52074, Germany

ARTICLE INFO

Keywords:
Process model comprehension
Business process management
Large language models
Generative AI

ABSTRACT

In Business Process Management (BPM), effectively comprehending process models is crucial yet poses significant challenges, particularly as organizations scale and processes become more complex. This paper introduces a novel framework utilizing the advanced capabilities of Large Language Models (LLMs) to enhance the comprehension of complex process models. We present different methods for abstracting business process models into a format accessible to LLMs, and we implement advanced prompting strategies specifically designed to optimize LLM performance within our framework. Additionally, we present a tool, AIPA, that implements our proposed framework and allows for conversational process querying. We evaluate our framework and tool through: i) an automatic evaluation comparing different LLMs, model abstractions, and prompting strategies; ii) a qualitative analysis assessing the ability to identify critical quality issues in process models; and iii) a user study designed to assess AIPA's effectiveness comprehensively. Results demonstrate our framework's ability to improve the comprehension and understanding of process models, pioneering new pathways for integrating AI technologies into the BPM field.

1. Introduction

Business Process Management (BPM) represents a management approach focusing on aligning an organization's processes with its strategic objectives. This includes process documentation, automation, integration, and continuous process improvement. Using BPM allows organizations to optimize performance, manage change, and achieve operational excellence [1].

In the context of BPM, *process models* serve as the primary artifact for depicting the flow of activities, data, events, and organizational units in a process. Process models facilitate the analysis, simulation, optimization, and automation of business processes. In today's competitive market, high-quality process models are pivotal for enterprises seeking to enhance their operational efficiency and the quality of their products and services. However, the inherent complexity of real-world business processes often results in intricate models that can be difficult to understand and manage. This complexity can lead to higher costs and more errors during the maintenance and improvement of the processes.

The Business Process Model and Notation (BPMN) [2] has become the de facto standard for modeling business processes and is widely used in industry. While BPMN offers a diverse range of elements and constructs, typical usage in industry centers around a limited, commonly used subset [3]. However, the availability of additional, more advanced elements allows for the modeling of specialized or complex scenarios. This capability enhances BPMN's versatility but also complicate model comprehension, increasing the risk of cognitive overload for users [4,5]. This complexity acts as a barrier, hindering users from fully comprehending BPMN models.

In addressing the challenge of comprehending complex process models, leveraging new technologies in AI hold promise. Among these technologies, *Large Language Models (LLMs)* stand out for their advanced capabilities in natural language processing and pattern recognition. Advanced LLMs, such as *gpt-4* are trained on vast amounts of text data, enabling them to generate human-like text and engage in natural language conversations. Due to their comprehensive training data, LLMs contain a wealth of process-related domain knowledge that can facilitate process model comprehension. Moreover, LLMs possess reasoning capabilities, allowing them to recognize, analyze, and make

☆ This article is part of a Special issue entitled: 'Generative AI' published in Decision Support Systems.

* Corresponding author at: Fraunhofer Institute for Applied Information Technology FIT, Schloss Birlinghoven, Sankt Augustin, 53757, Germany.
E-mail address: humam.kourani@fit.fraunhofer.de (H. Kourani).

https://doi.org/10.1016/j.dss.2025.114563

Available online 4 November 2025

inferences from textual data in various contexts [6]. These reasoning capabilities enable LLMs to comprehend process models, identify relationships between elements, and interpret process structures accurately.

In this paper, we present a novel framework that leverages the capabilities of LLMs to enhance the comprehension of complex BPMN models. By abstracting process models into different formats and employing advanced prompt engineering techniques, we guide LLMs to comprehend the different process structures and relationships. Our framework enables users to interact with the LLMs, gaining deep insights into complex processes without requiring technical expertise in modeling languages. Furthermore, we present our tool, AIPA (AI-Powered Process Analyst), which integrates our framework with state-of-the-art LLMs. Our research is structured around the following key research questions:

- RQ1: How does the choice of abstraction method for BPMN models affect their comprehension by LLMs?
- RQ2: What prompting strategies can enhance the comprehension capabilities of LLMs for BPMN models?
- RQ3: How effectively can state-of-the-art LLMs understand and interpret BPMN models?
- RQ4: How effective are LLMs at identifying structural and semantic flaws in BPMN models?
- RQ5: How does the application of LLMs influence user comprehension of BPMN models in practical scenarios?

The remainder of the paper is structured as follows. Section 2 discusses related work. In Section 3, we introduce our framework for LLM-based process model comprehension. In Section 4, we present our tool, AIPA. We evaluate our framework on different textual abstractions and prompting strategies in Section 5, and we present a qualitative analysis of the LLMs' quality assessment capabilities in Section 6. We then conduct a user study to assess the effectiveness and usability of AIPA in Section 7. Section 8 discusses the limitations of our framework and proposes ideas for future work. Finally, Section 9 concludes this paper.

2. Related work

The complexity of comprehending business process models has been extensively explored. In [7], the authors conducted a comprehensive review of metrics for assessing business process complexity, identifying different dimensions of complexity. In [8], the authors empirically evaluated the cognitive difficulty associated with comprehending process models, focusing on specific process constructs. Their findings revealed that repetition and exclusive choices impose a higher cognitive load compared to concurrent and sequential tasks. The study [9] assessed the comprehension of BPMN models by healthcare associates, highlighting the challenges they face in understanding complex BPMN models. These studies underscore the need for new methods to enhance the comprehension of process models.

Process model querying methods [10] play a crucial role in enhancing the accessibility and usability of business process models. These methods can be categorized into two main categories: *model-specific querying* [11–14] and *models repository querying* [15,16] (i.e., finding the models in a repository satisfying some given constraint). All the proposed approaches allow for a large number of queries. However, the model-specific approaches are limited by (i) lack of domain knowledge on the underlying process; (ii) prototypal implementation; (iii) the user needs to learn a new querying language (either graphical or textual). These limitations highlight the need for more intuitive and accessible querying methods, which our framework addresses by leveraging LLMs to facilitate conversational process querying.

AI systems can help non-expert analysts interpret BPMN models by automatically analyzing, extracting, and summarizing relevant information from process models [17]. In [18], the authors present a logical model for analyzing BPMN models using AI techniques. Their approach demonstrates the ability of AI systems to detect patterns, errors, and inconsistencies, thereby enhancing the analysis of BPMN models. In [19], the authors leverage conversational AI to create interactive systems that can understand and respond to natural language inputs, thus facilitating more intuitive and efficient management of business processes. These advancements underscore the potential of AI technologies to simplify the interpretation of complex process models.

Recently, some LLM-based model querying methods have been proposed. In [20], Petri nets are textually abstracted for process description and improvement purposes. Declarative models involving the control flow and time perspectives are textually abstracted in [21], allowing for a range of queries. In [22], a double-faced approach involving fine-tuning open-source LLMs and retrieval augmented generation techniques on a specific BPMN process model is proposed. However, the results of the assessment are not convincing, achieving low percentages on simple tasks such as tasks and flow existence. The approach described in [23] aims to explain decision points in a process exploiting both the structure of the process model and information extracted from the event log (for instance, causal relations and Shapley values). A limitation stated in the paper is that the implemented tool does not allow for iterative cycles, being limited to the initial response. Moreover, it is not possible to express questions unrelated to decision points.

3. LLM-based process model comprehension framework

This section introduces our framework for LLM-based process model comprehension. We outline the high-level architecture of the framework, and we propose several methods for abstracting process models and various prompting strategies aimed at optimizing LLM performance.

3.1. Architecture

Fig. 1 illustrates the architecture of our LLM-based process model comprehension framework. The process initiates with a user uploading a BPMN model, which is automatically abstracted into an input format that an LLM can process (cf. Section 3.2). For targeted analysis, users can optionally select specific parts of the BPMN model, and this selection is then also included in the abstraction to facilitate the analysis of specified areas of interest.

To aid the LLM in effectively understanding and analyzing the BPMN model, the model abstraction is augmented with tailored instructions. These incorporate various prompting strategies designed to optimize the LLM's response (cf. Section 3.3).

The user then poses an inquiry about the BPMN model, which is integrated into a prompt that combines the textual representation, the applied prompting strategies, and the user's inquiry. This enriched prompt is submitted to the LLM, which processes the input and generates a response. The response is then forwarded to the user. The interaction is dynamic, allowing the user to pose follow-up questions. Each follow-up question is integrated into a new prompt, maintaining the conversation history, and submitted to the LLM, which then generates subsequent responses.

3.2. BPMN abstraction

The abstraction of BPMN models is a crucial component of our framework, enabling their transformation into formats that can be effectively processed by LLMs. This section details the four abstraction formats supported by our framework: XML, simplified XML, JSON, and image.

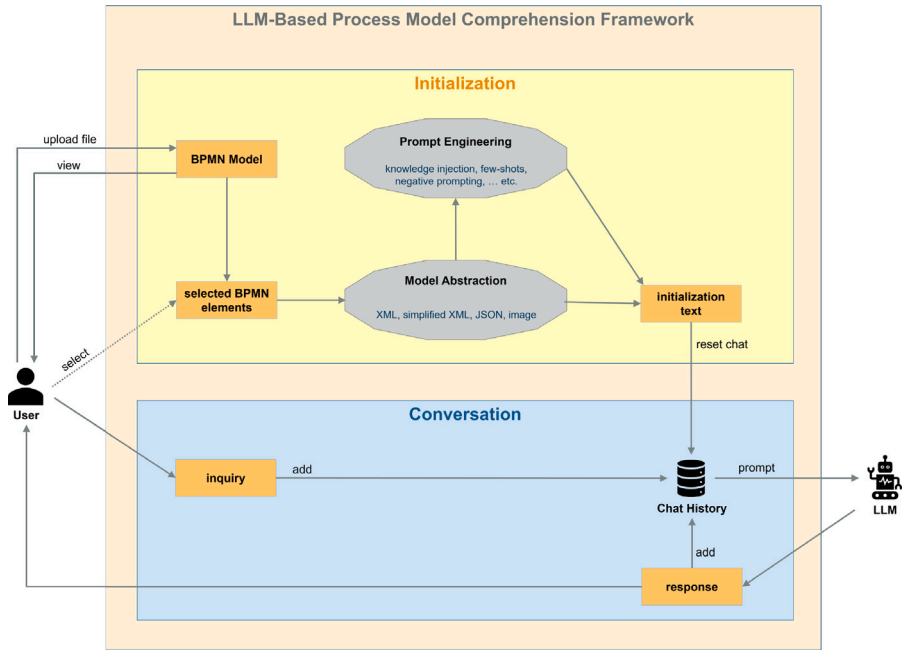

Fig. 1. LLM-based process model comprehension framework.

XML. BPMN Models of the 2.0 standard are typically stored in an XML format [2]. This comprehensive format encapsulates all visual and structural aspects of the process, including tasks, events, gateways, and detailed attributes such as positions, styles, and additional metadata. The Full XML abstraction retains all these details, conforming with BPMN 2.0 standard.

Simplified XML (SXML). We designed this abstraction format to reduce the complexity of the standard XML format by deliberately omitting non-essential elements such as layout information, styling, and metadata. These elements are considered irrelevant to the core logical structure of the BPMN model. SXML retains the original XML's hierarchical structure but includes only the fundamental components such as swimlanes, tasks, gateways, and connections. This abstraction provides a concise representation that highlights the operational aspects of the BPMN model without the additional complexity of visual details. Listing 1 shows the simplified XML abstraction of an example BPMN model.

Listing 1 Simplified XML abstraction for an example BPMN model.

```
<definitions Definitions_1>
   <collaboration col_1>
      <participant par_1> (My Process)
        - processRef: pro_1
      </participant>
   </collaboration>
   <process pro_1>
      <laneSet>
         <lane lane_1> (My Resource)
            <flowNodeRef (task_1)/>
            <flowNodeRef (event_1)/>
         </lane>
      </laneSet>
      <task task_1 (Task 1)/>
      <startEvent event_1 (Start)/>
      <sequenceFlow flow_1>
        - sourceRef: event_1
        - targetRef: task_1
      </sequenceFlow>
   </process>
</definitions>
```

JSON. We designed the JSON abstraction to restructure the BPMN data into an attribute-based representation. Unlike the hierarchical structure of XML, the JSON format organizes BPMN elements into a flat list, where each element is associated with a set of attributes that encapsulate all necessary information. Similarly to the Simplified XML abstraction, the JSON abstraction retains only the essential components and excludes non-essential attributes like styling and layout information. By doing so, it enables LLMs to process the structural and operational aspects of the process model efficiently, focusing on the attributes that are most relevant for analysis. Listing 2 shows the JSON abstraction of the same BPMN model from Listing 1.

Image (PNG). Recent advancements in LLMs have extended their capabilities beyond text processing to include support for image inputs. This enhancement allows for the processing and analysis of non-textual data, offering a broader range of applications. We leverage these capabilities through the image abstraction of BPMN models. This abstraction involves transforming the graphical BPMN models into PNG images. This format captures the visual layout of the model, preserving the spatial arrangement of the different BPMN elements. By feeding the generated image into an advanced model, its image processing capabilities are used to interpret the BPMN model's content and structure.

3.3. Prompt engineering techniques

Prompt engineering refers to the techniques used to instruct LLMs and guide them towards desired outputs. This includes providing additional information in the prompt to better inform the LLM about the requirements of the task and the domain knowledge required to perform it. The goal is to optimize the LLM's performance and improve the relevance and quality of its outputs.

In our framework, we support several prompting techniques that can be combined to optimize the understanding of BPMN models by LLMs. This section outlines these techniques, and the full prompts are available at https://zenodo.org/records/16017304.

Role prompting

One problem of LLMs is their *laziness*, which was empirically studied in [24]. Laziness should be interpreted as the tendency to reduce the effort to accomplish a task and can be either observed in generic answers, statements without explanations, or missing important parts of the original prompt. Some strategies have been proposed to help mitigate laziness. One strategy that can be used to mitigate this issue is

Listing 2 JSON abstraction for an example BPMN model.

```
- { $type: bpmn:Collaboration, id: col_1, $parent: Definitions_1 }
- { $type: bpmn:Participant, id: par_1, name: My Process, processRef: pro_1, $parent: col_1 }
- { $type: bpmn:Task, id: task_1, name: Task 1, lanes: (lane_1), $parent: pro_1 }
- { $type: bpmn:StartEvent, id: event_1, name: Start, lanes: (lane_1), $parent: pro_1 }
- { $type: bpmn:SequenceFlow, id: flow_1, sourceRef: event_1, targetRef: task_1, $parent: pro_1 }
- { $type: bpmn:Lane, id: lane_1, name: My Resource, flowNodeRef: (task_1, event_1) }
```

Listing 3 Implementation of role prompting.

```
- Your role: You are an expert in business process modeling and the BPMN 2.0 standard. I will give you a textual representation of a full BPMN
  model or only selected elements of a BPMN (e.g., a set of tasks and flows) and ask you questions about the process. You are supposed to
  answer the questions based on your understanding of the provided model.
- Please take the role of a process expert who is familiar with the domain of the provided process, and use your domain knowledge to better
  understand and analyze the process, filling in any missing gaps.
```

Listing 4 Instructing the LLM to use non-technical language.

```
- Please answer in natural language, so that any user not familiar with the BPMN standard can understand your answer without any technical
  knowledge; i.e., avoid technical terms like Task, Gate, flow, lane, etc.; rather use natural language to describe the behavior of the
  underlying process.
```

Listing 5 Implementation of chain of thoughts.

```
- Where possible (especially for complex queries), please share the chain of thoughts or reasoning behind your answers. This helps in
  understanding how you arrived at your conclusion.
```

Role prompting [25]. This strategy involves configuring the prompt to position the LLM as a domain expert [26]. As illustrated in Listing 3, our framework includes two implementations of role prompting:

- *Process Modeling Expert (S1):* We assign the LLM the role of an expert in business process modeling and the BPMN 2.0 standard.
- *Process Owner (S2):* We assign the LLM the role of a process expert familiar with the domain of the provided process, instructing it to use its domain knowledge to fill in any missing gaps when analyzing the process.

Non-technical abstraction (S3)

As shown in Listing 4, we instruct the LLM to avoid technical terms and use natural language to describe the behavior of the underlying process. This technique ensures that explanations and analyses are accessible to users who may not be familiar with the specialized terminologies of the BPMN 2.0 standard.

Chain of thoughts (S4)

LLMs are subject to problems such as *hallucinations* (the answer is partly unrelated to the original question) and *non-determinism* (the answer changes in different sessions). The chain-of-thoughts technique aims to improve the interpretability and reasoning capabilities of the LLM by explicitly requesting the motivations and intermediate reasoning steps it employs to answer the question [27]. Listing 5 illustrates our implementation of the chain of thoughts method.

Knowledge injection (S5)

LLMs are trained on a vast corpus of knowledge, but they might still miss context-specific information. To address this issue, *fine-tuning* techniques [28] revise the parameters of LLMs to include additional information about the task in question. However, due to the high computational costs associated with fine-tuning, an alternative is to engineer the prompts to include additional information. This strategy, known as *knowledge injection* [29], involves enriching prompts with task-specific information that the LLM may not have encountered during its initial training.

The BPMN standard is inherently complex, encompassing a wide range of elements and constructs. To balance the comprehensiveness of the information with the constraints of the LLM's context limit, we provide an informative summary of BPMN essential elements. The injected knowledge covers the following BPMN elements:

- Flow objects: events, gateways, tasks, sub-processes, transactions, and call activities.
- Connections: sequence flows, message flows, and associations.
- Further elements: pools, lanes, data objects, groups, and annotations.

Few-shot learning (S6)

This method leverages the LLM's ability to learn from limited examples by providing several pairs of example inputs and their expected outputs [30]. This choice over *zero-shot*, *one-shot*, and *many-shot* learning approaches stems from our goal to optimize the balance between model training efficiency and output accuracy. Zero-shot learning, where the model generates answers without prior related examples, often leads to less accurate or generalized responses due to the absence of context-specific training. One-shot learning provides a single example, which may not sufficiently capture the complexity or variability of the task. Many-shot learning, though potentially more effective due to a broader range of examples, requires a substantially larger dataset.

Given the constraints of LLM prompt size and processing capacity, we elected to utilize only one BPMN model for training, complemented with five pairs of question-and-answer examples. While we anticipate that including a broader array of examples across more models might yield more robust capabilities in BPMN comprehension, the current setup was chosen to maintain a reasonable trade-off between the prompt size and the richness of the training data. This strategic decision allows us to experiment within a feasible framework while setting the stage for potentially scaling up the number of examples or integrating more diverse models in the future to enhance the LLM's performance.

Listing 6 Examples used for few-shot learning. Lines are abbreviated with "..." for compactness.

```
- Let us consider the following textual representation of an example process: - { $type: ...
- These are example pairs of input and expected output:
Input 0: How does the bank start the credit scoring process?
Expected output 0: A bank clerk uses their software to request a credit score for a customer, ...
Input 1: What happens after the bank sends a scoring request to the agency?
Expected output 1: The agency performs an initial credit scoring, and if this initial scoring ...
Input 2: What occurs if the initial credit scoring does not give an immediate result?
Expected output 2: The agency informs the bank's system about the delay and starts a more ...
Input 3: How is the clerk informed of the final credit scoring result?
Expected output 3: Once the scoring is completed and the result is sent back to the bank's ...
Input 4: What does the bank do if the credit score is delayed?
Expected output 4: The bank's system sends a notification to the clerk to report the delay, ...
```

Listing 7 Extending the few-shot pairs by including examples of undesired outputs. Lines are abbreviated with "..." for compactness.

```
- Now, I will give you example bad outputs for the same example questions, so you try to avoid ...
Bad output for question 0: The bank initiates a BPMN Collaboration, invoking a Participant ...
Bad output for question 1: Upon receipt of the scoring request, an IntermediateCatchEvent is ...
Bad output for question 2: If the initial scoring does not resolve, a conditional ...
Bad output for question 3: The final scoring outcome, after processing through a sequence of ...
Bad output for question 4: In the event of a scoring delay, an IntermediateCatchEvent captures ...
```

The BPMN model we use for implementing few-shot learning is the credit scoring process, available at https://github.com/camunda/bpmn-for-research. We crafted a set of five questions designed to challenge the LLM across various aspects of the process: starting the process, handling immediate outcomes, dealing with delays, and finalizing results. Moreover, the questions target different BPMN elements: tasks, events, pools, gateways, and message flows. The provided expected outputs for these questions were designed to not only provide accurate responses but also to use plain, accessible language that enhances comprehension for users without technical knowledge. Listing 6 illustrates our few-shot learning example pairs.

Negative prompting (S7)

Negative prompting involves guiding the LLM by explicitly stating what should be avoided in its responses [31]. We enhance our few-shot learning demonstrations by including examples of undesired outputs as shown in Listing 7. We generated these responses by providing a textual abstraction of the BPMN model and the questions to GPT-3.5 without implementing any prompting strategies. The resulting answers include inaccuracies and misleading information. Additionally, they tend to be overly complex, filled with unnecessary technical details that obscure the clarity and relevance of the responses.

4. Tool support

In this section, we present our tool AIPA (AI-Powered Process Analyst) which integrates our LLM-based process model comprehension framework, as detailed in Section 3, with a variety of state-of-the-art LLMs. This includes Google, OpenAI, DeepSeek, Anthropic, Deepinfra, Mistral AI, OpenRouter, Cohere, and Grok. This integration is flexible, allowing users to configure the tool to use any LLM available through the supported AI providers. The tool can be downloaded from https://zenodo.org/records/16017304 and used following the instructions contained in the **README.md** file. Additionally, AIPA is accessible online at https://www.processquerying.app/.

AIPA uses SXML as the default textual abstraction format to represent BPMN models. This choice was informed by our evaluation experiments in Section 5, which show that the JSON and SXML formats provided a clearer and more effective simplification compared to the other abstraction formats we proposed in Section 3.2. The prompting strategies proposed in Section 3.3 are all implemented in AIPA and enabled by default.

Fig. 2 shows a screenshot of AIPA. The user can upload a BPMN model and start a chat about the uploaded model with an AI assistant. When users select specific elements within a BPMN model, the selected parts are also included in the generated prompt. Additionally, our tool supports voice interactions, allowing users to input their questions and receive responses audibly. The dynamic nature of the tool facilitates an interactive dialogue between the user and the LLM. Users can pose follow-up questions, maintaining the conversation history. Users can reset the conversation at any time to start a new chat.

Note that before starting the conversation with the AI assistant, the user needs to configure the desired AI provider connection by selecting an LLM and entering the corresponding API key.

5. Evaluation

In this section, we comprehensively evaluate our framework by addressing the first three research questions we defined in Section 1:

- **Abstraction Methods (RQ1):** In Section 5.2.1, we compare the different BPMN abstraction methods supported within our framework.
- **Prompting Strategies (RQ2):** In Section 5.2.2, we explore the effect of various prompting strategies on the comprehension of BPMN models by LLMs.
- **State-Of-The-Art LLMs (RQ3):** In Section 5.2.3, we assess the overall effectiveness of integrating state-of-the-art LLMs with our framework.

5.1. Setup

This section describes the experimental setup. First, we detail the dataset comprising diverse BPMN models and user queries in Section 5.1.1. Then, we outline our scoring mechanism used to evaluate LLM outputs in Section 5.1.2.

5.1.1. Dataset

We use three BPMN models in our experiment. First, we use a publicly available BPMN model (*Healthcare Process*) [32]. This model represents a standard medical process and is publicly available with a textual description. It provides a baseline for comparison but is limited to simple workflow elements.

To ensure an unbiased evaluation and to address the possibility that the healthcare model and its description might have been included in the training data of the LLMs, we designed two additional BPMN models specifically for this study. These models incorporate a broader range

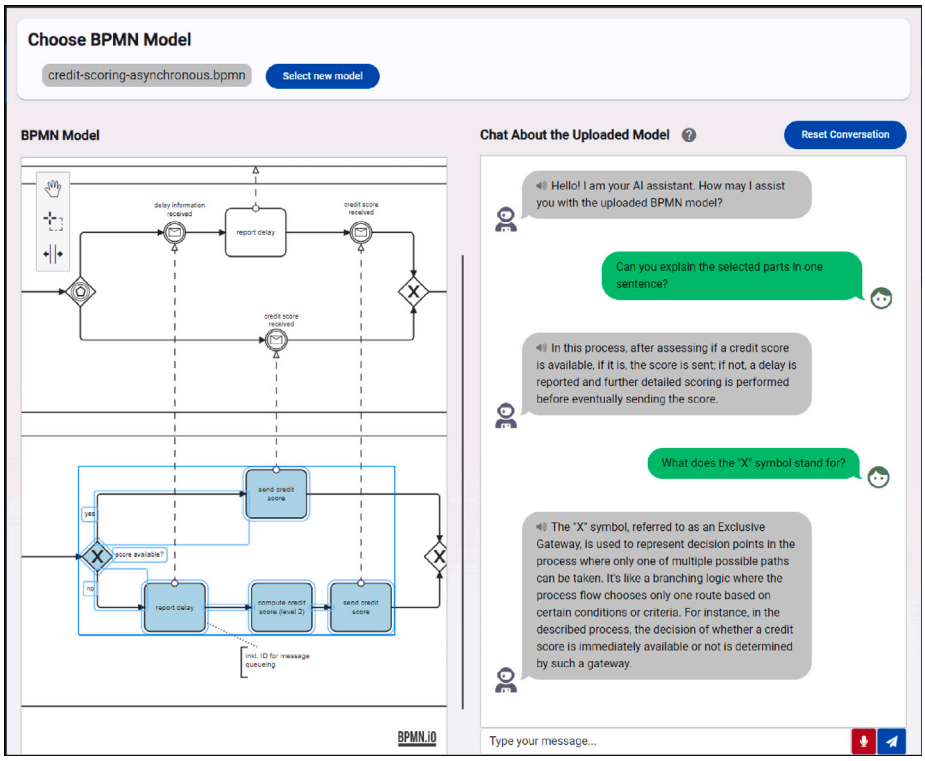

Fig. 2. Screenshot of AIPA.

of BPMN elements such as events, data objects, and complex gateways, ensuring a comprehensive assessment of the LLMs' capabilities across various BPMN facets.

The following list gives an overview of the two additional BPMN models we designed for our evaluation together with the help of BPMN experts from our research team:

- *Dispatch of Goods*: This process, shown in Fig. 3, details the preparation and dispatch of goods, starting from the decision on shipping methods to the final packaging and shipment preparation. It includes advanced BPMN elements such as time events, data objects, and inclusive gateways.
- *Order Manufacturing*: This process, shown in Fig. 4, revolves around handling and fulfilling customer orders within a company, engaging multiple departments from sales to warehouse. It captures various advanced BPMN elements such as conditional flows, events (e.g., messages, compensation), and sub-processes.

To thoroughly evaluate the quality of answers generated by the LLMs, we generate questions that cover the various aspects shown in Table 1. These aspects were designed to cover different facets of process understanding, ranging from basic control-flow sequencing to complex data interactions and organizational roles. However, it is important to note that for the healthcare process, we do not have questions that capture all of these aspects as the corresponding model lacks advanced BPMN elements. The questions for the three processes are shown in Tables 2, 3, and 4.

For the healthcare process, we utilize the overall textual model description of the underlying process as ground truth. For the other two models, we designed a ground truth answer for each question.

Table 1
Types of questions.

	Type	Description
A1	Open-Ended	Questions allowing for expansive, subjective responses rather than strict factual answers.
A2	Close-Ended	Questions prompting definitive answers, such as "yes" or "no", or leading to specific factual information.
A3	Control-Flow	Questions concerning the relationships of tasks and gateways.
A4	Data-Perspective	Questions related to the relationships between data objects and other artifacts.
A5	Notation	Questions related to the syntax of the BPMN elements in the context of the given process model.
A6	Domain-Specific	Questions specific to a domain (e.g., finance, healthcare, manufacturing).
A7	Organizational-Specific	Questions specific to the practice or policies of an organization.
A8	Role-Perspective	Questions related to involved actors and their responsibilities.
A9	Event Relationships	Questions concerning relationships between events and other artifacts.

5.1.2. Outputs scoring

In [33], evaluation criteria for LLMs' outputs on process mining tasks are proposed. These criteria include *human evaluation*, where a human evaluates the text provided by the LLM; *automatic evaluation*, which is only applicable to quantitative questions; and *self-evaluation*, where LLMs evaluate LLM outputs. Particularly, *self-reflection* methods, where the output of one session is provided to another LLM or another

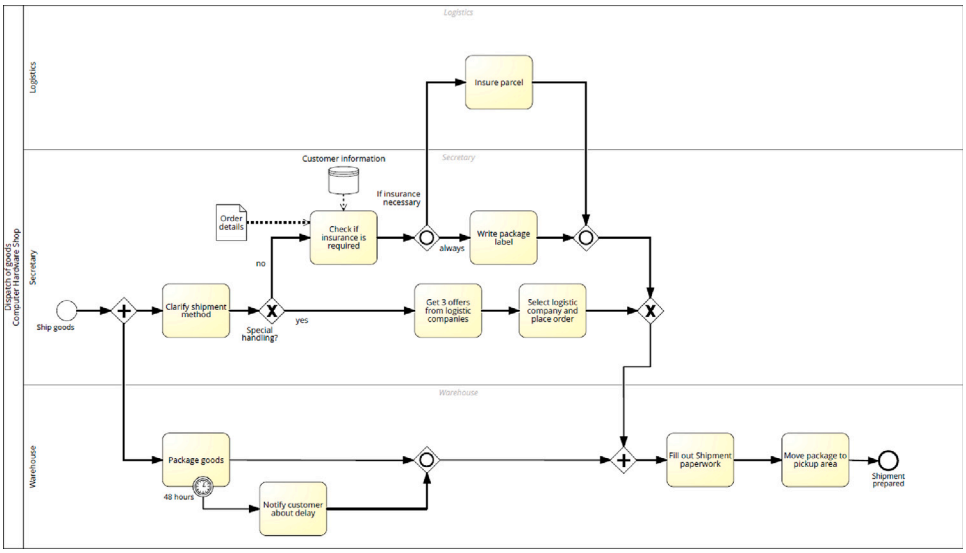

Fig. 3. Dispatch Of Goods BPMN model.

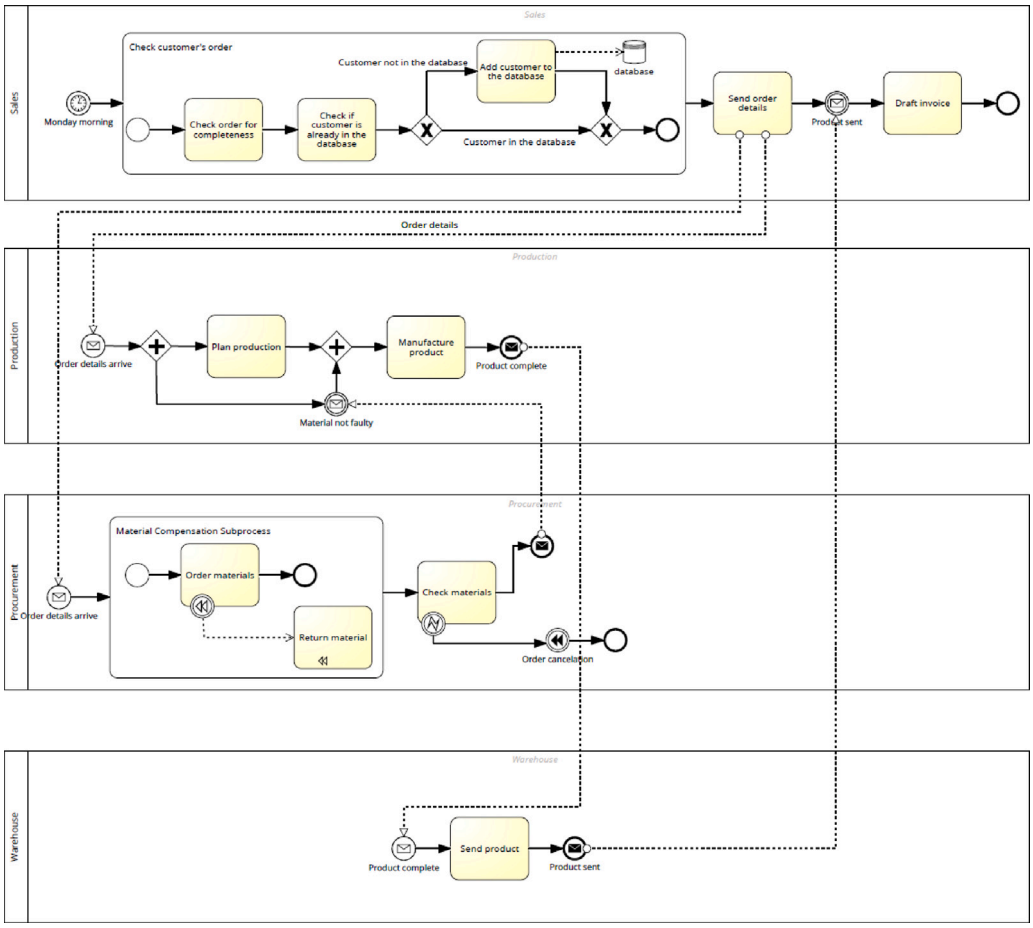

Fig. 4. Order Manufacturing BPMN model.

session of the same LLM to score the quality of the output, can be employed for self-evaluation [34].

For the evaluation of our framework, we decided to employ LLM self-evaluation due to the large size of the experiments. We incorporate a direct comparison of LLM outputs to ground truth answers. We utilize *gpt-4o* to perform a self-evaluation, assigning quality scores to LLM responses based on their alignment with the ground truth answers. To assess the robustness of our evaluation, we conducted a detailed review involving human experts; specifically, we assigned this role to the team who designed the models and the ground truth answers. We selected a sample encompassing two questions from each process, with four different LLM-generated answers for each question. Our experts

Table 2
Inquiries used for the "Healthcare" process.

	Question
1	What are the initial steps taken to prepare for the procedure?
2	Why is hand washing crucial before proceeding with the operation?
3	At what stage of the process is the puncture area cleaned?
4	What is the significance of putting sterile gel before the procedure?
5	How does ultrasound configuration play into the overall procedure?
6	What techniques are used to identify the correct anatomical puncture site?
7	At what point is the decision made regarding the need for anesthetics?
8	How is the guidewire used in the procedure, and what are its implications?
9	Is there a verification step to ensure the wire and catheters are correctly positioned?
10	How do operators decide to widen the pathway for the catheter?
11	What measures are taken to confirm the flow and reflow in the catheter?
12	What steps are involved in the removal of the guidewire?
13	How is the final position of the catheter checked before concluding the procedure?
14	What are the potential risks if the catheter position is not correctly verified?
15	Why might there be a need for further intervention after advancing the catheter?
16	How does the process ensure the sterility and safety of the procedure throughout?
17	What role does Doppler and compression identification play in the procedure?
18	How do practitioners decide between using Doppler identification, anatomic identification, and compression identification techniques?
19	After how many stages is blood return expected, and why is it important?
20	What are the concluding steps of the procedure, and how is success determined?

were also provided with the rationale used by *gpt-4o* for scoring these answers, to assess both the quality and fairness of the evaluation process.

The feedback we received on the LLM self-evaluation has been positive in general, affirming that the quality of the evaluation is high and the reasoning behind the scores is both transparent and well articulated. Although there is an inherent level of subjectivity in any evaluation, our experts agreed that the scores assigned by *gpt-4o* were justified and in line with their understanding of the content. The detailed and contextually rich responses from the LLMs were particularly praised for their relevance. An oversight in detecting an error in one answer was noted; however, it did not detract significantly from the overall success of the evaluation method. In general, our experts confirmed the reliability and high accuracy of the LLM self-evaluation approach.

5.2. Results

This section presents the results of our automatic evaluation. All scripts, data, and results are available at https://zenodo.org/records/16017304.

5.2.1. Evaluating abstraction formats

In this section, we perform a comparative evaluation of the different abstraction formats defined in Section 3.2. We configure our framework to use *gpt-4o-2024-05-13*, and we only enable simple prompting strategies (S1, S2, S3, and S4) that are efficient in terms of token usage, allowing us to focus on the impact of the abstraction format itself on the comprehension of BPMN models. The results of evaluating the effect of the abstraction formats on LLM comprehension are detailed in Table 5, where we report the average quality scores across various question categories, the overall average quality score, and the number of tokens required for each format.

Our findings indicate that JSON, simplified XML (SXML), and XML generally produce similar comprehension scores, with PNG lagging behind overall. This difference becomes especially apparent for the healthcare process, where the PNG format scored very low compared to the other formats. However, PNG showed better performance in answering data-related and role-perspective questions, likely due to its visual nature, which makes artifacts and swimlanes more recognizable.

XML stands out for the Order Manufacturing process. Its detailed representation leads to superior results across all categories but at the cost of a significantly higher token consumption compared to JSON and SXML.

In summary, the choice of the abstraction format depends on the type of question and the complexity of the processes involved. While XML and PNG each excel in certain scenarios, JSON and SXML consistently offer the best trade-off between quality and token consumption.

5.2.2. Evaluating prompting strategies

In this subsection, we explore the impact of the prompting strategies we defined in Section 3.3 on LLMs' comprehension of BPMN models. We employing both the JSON and simplified XML abstraction formats for each BPMN model, and we configure our framework to use *gpt-4o-2024-05-13*. Each prompting strategy is assessed individually to determine its specific effect; however, for practical applications, combining these strategies is recommended to achieve optimal results. The average quality scores and standard deviation values for each strategy are reported in Table 6.

The results show that strategies S3 (non-technical restriction), S6 (few-shot learning), and S7 (negative prompting) stand out with particularly strong positive impacts across the different models and abstraction formats. For S3, restricting LLMs to use only natural language significantly enhances model comprehension, likely due to improved readability and accessibility of the information. Strategies S6 and S7 also align with expectations, showing a positive effect and confirming the usefulness of both positive and negative example-based learning for enhancing LLM comprehension.

Conversely, strategies S1 and S2, which involve role prompting, demonstrate minimal impact on the comprehension capabilities of LLMs. This outcome suggests that simple identity-based prompts do not significantly influence LLM performance in this context. Similarly, the chain-of-thoughts approach (S4) and the knowledge injection strategy (S5) show limited effects. For S4, the tendency to generate unnecessarily lengthy outputs might overwhelm the user with information that does not directly aid in model comprehension. For S5, the lack of observed improvement might be attributed to the extensive availability of information on the BPMN 2.0 standard online, which the LLM could have already encountered during its training. Future work could explore more targeted knowledge injection strategies, focusing specifically on the complex elements of the given BPMN model rather than providing general knowledge.

Table 3
Inquiries for the "Dispatch of Goods" process.

	Question	A1	A2	A3	A4	A5	A6	A7	A8	A9
1	Can the activities "package goods" and "get 3 offers from logistic companies" be executed in parallel?		x	x						
2	Give an explanation for the process steps that the secretary carries out!	x		x	x		x		x	
3	Give an explanation for the "warehouse" process!	x		x			x		x	x
4	How many activities does Logistics perform?		x							
5	What does the gateway after the activity "clarify shipment method" mean?	x		x		x				
6	When does the logistics department need to insure parcel?		x	x				x		
7	How does the process end?		x	x						
8	How many organizational units are involved in the process?		x						x	
9	Can writing the package label be executed simultaneously to selecting a logistic company and placing an order?		x	x				x		
10	What data does the secretary use when it checks if an insurance is required?		x		x			x		
11	When does the warehouse have to notify the customer about the delay?		x	x				x		x
12	Is the activity "package goods" finished if 48 h have passed?		x	x						x
13	Can the activities "insure parcel" and "write package label" be executed simultaneously?		x	x						
14	Is it possible that only one of the activities "insure parcel" and "write package label" is executed?		x	x						
15	Which activity includes information retrieval from a database?		x		x					

5.2.3. Evaluating state-of-the-art LLMs

In this section, we evaluate the performance of state-of-the-art LLMs on our framework. We configure our framework to use the Simplified XML (SXML) abstraction, and we enable the first four prompting strategies (S1, S2, S3, and S4). We have selected four LLMs for this analysis:

- **GPT-4o-2024-05-13**: The latest iteration of OpenAI's language models at the time of the experiment.[1]
- **GPT-4-Turbo-2024-04-09**: A variant of GPT-4 optimized for speed and efficiency, potentially sacrificing some depth in exchange for faster response times.
- **Microsoft WizardLM-2-8x22B**[2]: An open-source high-capacity model from Microsoft, designed to excel in deep contextual understanding and complex reasoning tasks.
- **Mixtral-8x22B-Instruct v0.1**[3]: An open-source instructive model geared towards following explicit user instructions with high precision, using fewer tokens for outputs which might affect the depth of generated content.

We also considered smaller LLMs (**Mixtral-8x7B-Instruct v0.1**, **Codestral**, **Mistral 7B**) in our initial experiments, but in light of significantly worse results, we discarded them. Some other LLMs were discarded for their limited context window (**Llama3-70B**, **Nemotron-4-340B-Instruct**).

The results of our comparative analysis are shown in Table 7, where we report the average quality scores for different question categories along with the number of output tokens produced by each LLM.

All LLMs demonstrated similar levels of performance; however, **GPT-4o-2024-05-13** stands out in domain-specific questions, possibly due to its more extensive training on a wider range of topics and scenarios. This suggests that a broader training corpus can significantly benefit domain-specific comprehension.

Notably, the results show that open-source LLMs perform comparably to commercial alternatives, highlighting the advancements and accessibility in LLM technology.

Interestingly, there is a consistent trend where increased token usage correlates with better scores. This suggests that more detailed responses, which utilize more tokens, tend to provide richer and more useful information, thereby improving the model's score in our evaluation. Among the evaluated models, **Mixtral-8x22B-Instruct** uses the fewest tokens but maintains competitive performance, indicating efficiency in token usage. However, the detailed, token-rich outputs from models like **GPT-4o** and **WizardLM** indicate that when the depth of description is critical, especially for complex topics, higher token consumption may be beneficial.

6. Qualitative analysis of process model quality assessment

To extend our evaluation beyond the comprehension of well-structured processes, we conducted a qualitative experiment designed to assess the framework's capabilities on imperfect process models. This analysis addresses RQ4, investigating whether LLMs can perform a basic level of quality assurance to identify both critical structural flaws and semantic inaccuracies in BPMN models.

[1] https://openai.com/
[2] https://ollama.com/library/wizardlm2:8x22b
[3] https://ollama.com/library/mixtral:8x22b-instruct-v0.1-fp16

Table 4
Inquiries for the "Order Manufacturing" process.

	Question	A1	A2	A3	A4	A5	A6	A7	A8	A9
1	How does the sales process work?	x		x	x		x		x	
2	What is the condition for the production to manufacture the product?		x						x	x
3	How does the subprocess in the activity "check customer's order" work?	x		x	x			x		
4	Who is responsible for sending the product to the customer?		x					x	x	
5	What happens after the order details arrive at the procurement team?	x		x			x		x	x
6	What happens if an error occurs when checking the materials?	x		x				x		x
7	How does a compensation work?	x		x						x
8	Why does the activity "return material" contain a compensation symbol?		x			x				
9	Why does the compensation need its own subprocess?	x		x		x				x
10	What is the meaning of a message symbol inside of an event?		x			x				x
11	What is the difference between a filled event symbol and a non-filled event symbol?		x			x				x
12	What does the parallel gateway mean?		x	x		x				
13	Is the material checked after manufacturing the product?		x				x			x
14	Does the plan production occur after manufacture product?		x	x			x			
15	Does the sales team check the customer's order?		x				x		x	
16	Is the product sent before drafting the invoice?		x	x						

Table 5
Effect of abstraction formats on the comprehension of BPMN models by *gpt-4o-2024-05-13*. Average quality scores and numbers of required tokens are reported.

(a) Healthcare process

	JSON		SXML		XML		PNG	
	Score	#Tokens	Score	#Tokens	Score	#Tokens	Score	#Tokens
All	8.5 ± 0.8	8918 ± 2	**8.7 ± 0.6**	7079 ± 2	8.5 ± 1.1	26329 ± 2	4.9 ± 2.5	1015 ± 2

(b) Dispatch of goods

Question Group	JSON		SXML		XML		PNG	
	Score	#Tokens	Score	#Tokens	Score	#Tokens	Score	#Tokens
A1	**8**	4542	**8**	3829	7.83	16 509	7.5	1353
A2	6.69	4544.08	**6.73**	3831.08	6.42	16 511.08	6.42	1355.08
A3	**7.58**	4544.17	7.54	3831.17	7.13	16 511.17	6.54	1355.17
A4	7.33	4541.67	8	3828.67	7.83	16 508.67	**8.83**	1352.67
A5	**9**	4545	8.5	3832	**9**	16 512	8.5	1356
A6	7.5	4540.5	**7.75**	3827.5	7.25	16 507.5	7	1351.5
A7	6.25	4544	6	3831	6.25	16 511	**7.38**	1355
A8	**8**	4540.33	8	3827.33	7.83	16 507.33	7.33	1351.33
A9	7	4543.75	**7.38**	3830.75	6.63	16 510.75	6.75	1354.75
All	6.9 ± 2.1	4544 ± 5	**7.0 ± 2.0**	3830 ± 4	6.7 ± 2.8	16511 ± 5	6.6 ± 2.2	1355 ± 5

(c) Order manufacturing

Question Group	JSON		SXML		XML		PNG	
	Score	#Tokens	Score	#Tokens	Score	#Tokens	Score	#Tokens
A1	7.5	11 338.14	7.36	8170.14	**8.21**	28 365.14	8.07	1011.14
A2	8.45	11 338.8	7.9	8170.8	**8.85**	28 365.8	7.6	1011.8
A3	7.78	11 337.33	7.72	8169.33	**8.61**	28 364.33	8.22	1010.33
A4	7	11 338.5	7.75	8170.5	**8.25**	28 365.5	**8.25**	1011.5
A5	8.4	11 339.6	7.7	8171.6	**8.6**	28 366.6	7.9	1012.6
A6	8.08	11 337.67	7.58	8169.67	**8.5**	28 364.67	7.42	1010.67
A7	7.33	11 340	7.5	8172	**8.5**	28 367	8	1013
A8	7.67	11 338.5	7.67	8170.5	**8.67**	28 365.5	8.58	1011.5
A9	8.17	11 339.11	7.22	8171.11	**8.22**	28 366.11	7.06	1012.11
All	8.1 ± 0.9	11339 ± 3	7.7 ± 1.4	8171 ± 3	**8.6 ± 0.8**	28365 ± 3	7.8 ± 1.8	1012 ± 3

Table 6
Effect of prompting strategies on the comprehension of BPMN models by *gpt-4o-2024-05-13*. For each model, both the JSON and simplified XML abstractions are considered. The average and standard deviation values are reported.

Model	Abst.	None	S1	S2	S3	S4	S5	S6	S7
Healthcare	JSON	6.8 ± 2.3	6.3 ± 2.5	7.4 ± 1.9	**8.4 ± 0.9**	6.4 ± 2.2	7.4 ± 2.0	7.3 ± 2.1	**8.3 ± 1.1**
	SXML	7.4 ± 2.0	7.5 ± 1.9	7.5 ± 2.1	**8.7 ± 0.7**	7.1 ± 2.1	6.8 ± 2.5	7.0 ± 2.5	**8.4 ± 1.1**
Order	JSON	6.0 ± 2.1	6.2 ± 2.1	6.4 ± 1.5	**7.7 ± 1.4**	6.4 ± 1.8	6.2 ± 1.8	**6.9 ± 1.4**	**7.2 ± 1.6**
	SXML	5.8 ± 1.6	6.0 ± 1.6	5.6 ± 1.7	**8.1 ± 1.0**	6.3 ± 1.4	6.1 ± 1.9	**6.9 ± 1.6**	**7.1 ± 1.5**
Dispatch	JSON	6.0 ± 1.9	5.5 ± 1.9	5.9 ± 1.6	**6.6 ± 2.0**	5.6 ± 1.7	5.7 ± 1.8	**6.8 ± 2.0**	6.5 ± 2.1
	SXML	5.7 ± 1.3	6.1 ± 2.1	6.2 ± 1.6	**7.2 ± 1.8**	5.6 ± 1.5	5.1 ± 1.6	**6.7 ± 1.9**	5.7 ± 2.5

Table 7
Performance of state-of-the-art LLMs on the comprehension of BPMN models. Average quality scores and average numbers of output tokens are reported.

(a) Healthcare process

	gpt-4o		gpt-4-turbo		WizardLM-2-8 × 22B		Mixtral-8 × 22B-Instruct	
	Score	Output Tok.	Score	Output Tok.	Score	Output Tok.	Score	Output Tok.
All	**8.7 ± 0.6**	408 ± 168	8.6 ± 0.8	319 ± 108	8.5 ± 1.1	430 ± 110	8.3 ± 1.0	154 ± 94

(b) Dispatch of goods

Question Group	gpt-4o		gpt-4-turbo		WizardLM-2-8 × 22B		Mixtral-8 × 22B-Instruct	
	Score	Output Tok.	Score	Output Tok.	Score	Output Tok.	Score	Output Tok.
A1	**8**	504.33	7.5	414.33	7.67	448	7.33	222
A2	6.73	263.54	6.31	267.38	**7.31**	363.62	6.5	146.08
A3	7	352.25	6.54	347.08	**7.46**	415.67	6.58	173
A4	**8**	393	7.67	279.33	7.33	390.67	6.33	184.67
A5	**8.5**	330	8	200	8	320	8	135
A6	**7.75**	591.5	7.25	521.5	7.5	512	7	265.5
A7	6	246	5.5	219	**6.75**	411.75	6.5	148.25
A8	**8**	434.33	7	407.67	**8**	440.67	7.83	221.67
A9	**7.38**	366	7.13	374	7.25	428	6.63	199.75
All	7.0 ± 2.0	309 ± 160	6.5 ± 1.9	295 ± 172	**7.4 ± 1.2**	371 ± 110	6.7 ± 1.7	160 ± 86

(c) Order manufacturing

Question Group	gpt-4o		gpt-4-turbo		WizardLM-2-8 × 22B		Mixtral-8 × 22B-Instruct	
	Score	Output Tok.	Score	Output Tok.	Score	Output Tok.	Score	Output Tok.
A1	7.36	449.86	**7.64**	406.57	7	501.43	7.07	163.29
A2	**7.9**	330.1	**7.9**	266.3	7.75	345.1	7.1	130.4
A3	**7.72**	403.22	7.56	364.33	7.67	466.33	7.5	152.78
A4	7.75	512.5	7	411.5	**8.5**	475	8	158.5
A5	7.7	433.2	**8.2**	353.2	7.2	411.2	6.6	179.6
A6	**7.58**	354.33	**7.58**	308	7.25	412.17	7.08	116.5
A7	7.5	300.67	7.17	321	7.33	408	**7.67**	167.33
A8	7.67	347.67	**7.75**	322	7.33	358.5	7.08	126.83
A9	7.22	408.22	**7.89**	370.78	6.78	458.78	6.39	158.78
All	7.7 ± 1.4	379 ± 132	**7.8 ± 0.9**	324 ± 119	7.4 ± 1.5	409 ± 115	7.1 ± 1.5	144 ± 53

6.1. Experimental setup

For this experiment, we selected five diverse BPMN models publicly available at https://github.com/camunda/bpmn-for-research. We then queried four different state-of-the-art LLMs (*gemini-2.5-flash*, *gemini-2.5-pro*, *gpt-4.1-nano*, and *o4-mini*) using our framework with the SXML abstraction. To probe the LLMs' reasoning capabilities without leading them, we used a single, open-ended query:

"Can you identify any quality issues in the process model?"

The complete, verbatim responses from all LLMs for this experiment can be found at: https://zenodo.org/records/16017304. We evaluated the LLM responses based on three criteria:

- Issue Detection Rate: The number of correctly identified quality issues versus the total number we had identified in our own manual analysis.
- Explanation Quality: The accuracy and clarity of the LLM's reasoning when describing a detected issue.
- Absence of Hallucination: Whether the analysis was grounded in the provided model or if the LLM invented non-existent flaws or made factually incorrect statements.

Prior to the automated evaluation, we performed a manual analysis of the models to establish a benchmark for the LLMs' performance. We identified the key quality issues summarized in Table 8.

6.2. Results

The results, summarized in Table 9, highlight the significant potential of LLMs for automated process model quality assessment, but simultaneously raise concerns about the risks of fully relying on their output. The analysis of the Dispatch-of-goods model serves as an excellent example of this duality. Here, the lighter model, gemini-2.5-flash, successfully identified the critical soundness violation, explained it, and proposed how to fix it. However, the more advanced model gemini-2.5-pro and the specialized reasoning model o4-mini, while also locating the error, elaborated further by diagnosing its effect as a deadlock. This lowered the quality of their answers, as the lack of synchronization issue in this process model actually leads to a duplicate execution of the task "Prepare for picking up goods". Surprisingly, gpt-4.1-nano failed to detect this major flaw, instead providing only generic improvement suggestions. Furthermore, we observed clear instances of hallucination, where some LLMs would make factually incorrect claims, such as stating that "no messages existed between pools" when they were explicitly modeled.

Table 8
Identified process model issues.

Process model	Quality issues
Dispatch of goods	• Parallel flows are merged using an exclusive (XOR) gateway.
Self-service restaurant	• Deadlock: "Place order" and "Collect money" wait for messages from each other. • Deadlock and potential infinite loop: "Get meal" never receives its required message and "Call guest" is executed every five minutes with no way of exiting that loop. • An invalid message flow from the "Call guest" task to the "Guest" pool itself, which has no effect.
Recourse	• A semantic flaw where a reminder is sent immediately without a preceding wait period. • Unnecessary complexity, as the "close case" task and "case closed" end events are duplicated three times. • An ambiguous endpoint where the process can terminate with a "case open" status, which is atypical.
Credit scoring (Synchronous)	• A critical deadlock where the "request credit score" task waits for a message that can never be received.
Credit scoring (Asynchronous)	• Unnecessary complexity from the duplication of the "send credit score" task and its related events.

In summary, our experiment highlights a dual finding. On one hand, LLMs demonstrate a powerful, emergent capability to assist in the complex task of process model quality assessment. On the other, they may overlook certain issues, provide answers with semantic inaccuracies, or even include clear hallucinations. Therefore, while LLMs are invaluable tools that can significantly accelerate analysis and help users better understand process models, full reliance on their results is inadvisable.

7. User study

In this section, we present a user study designed to assess the usability and effectiveness of our LLM-based framework for process model comprehension in real-world contexts. The study specifically addresses RQ5 (cf. Section 1), exploring how industry experts in the field of business process management perceive and interact with BPMN models using our tool, AIPA. It is important to note that the version of AIPA used in this study did not include the voice support functionalities, as these features were added subsequently. The user study was conducted by configuring AIPA to use *gpt-4-1106-preview*.

7.1. Setup

We conducted a total of six interviews, each lasting between 45 min and one hour. The interview partners were contacted via email and selected for their extensive experience with BPMN modeling in professional contexts and their background knowledge of how LLMs work. The experts were identified through the professional networks of the authors, who are well-connected within the BPM research and industry community. The selected experts represent a diverse range of professional experiences, from 0–5 years to 15–20 years in the field. All invited experts agreed to participate in the user study. Table 10 provides an overview of the interview partners.

The user study comprised six semi-structured interviews guided by a 16-question interview guide. Conducting semi-structured expert interviews allowed us to focus on the research topic while gathering in-depth information [35].

The interview guide is divided into three sections. The first section consists of three questions aimed at understanding the interviewee's expertise and creating a comfortable interview atmosphere, as these questions are easy to answer.

Next, we presented AIPA, explained its functionality, and gave the interviewees enough time to familiarize themselves with the interface and the selected BPMN model. We chose a BPMN model of low complexity to facilitate quick understanding and interaction. Once they indicated that they had understood the model, we asked the interviewees to ask the AI assistant questions about the BPMN model displayed. On average, four questions were asked by each interviewee.

After the user had time to interact with the tool, we moved on to the second section of the interview guide. The first question focused on evaluating the user interface, while the remaining eleven questions were aimed at evaluating the responses. These eleven questions are structured using the eleven criteria summarized in cf. Table 11. For each question, the interviewees were first asked to provide a rating on a five-point scale about the degree of fulfillment of the criterion and then to explore the reasons for the rating. By incorporating these tested criteria, we have followed a robust evaluation framework that improves the functionality of our system and ensures its usability.

After answering the twelve questions from the second section, we concluded the interview with an open question that gave the interviewee the opportunity to address aspects not covered by the previous questions.

7.2. Results

This section provides a comprehensive summary of the interview results, offering insights and key findings derived from the discussions.

User-friendliness. The user-friendliness aspect of the tool is highly praised as it is clear and concise, making it easy for users to interact with the system (Experts 1, 4). Users appreciate that the BPMN model is visible and can be reset, but they suggest further improvements such as displaying multiple process models at the same time to be able to simultaneously analyze several business processes that are connected with each other and improving visual elements, such as adapting the size of the window that contains the model (Expert 3). The ability to drag elements of the BPMN model directly on the user interface is highlighted as positive (Expert 5). It is suggested to further integrate the possibility of enlarging the model view to allow better interaction (Experts 5, 6) and possibly implement a function that highlights in color which part of the model the AI assistant's response refers to (Expert 6).

Soundness (C1). Regarding the question of soundness, the explanations provided by the AI assistant were generally effective and answered most questions well, although sometimes the answers could have been more precise (Experts 1, 3, 4). The AI assistant explained elements such as message flow accurately, although its impact on the workflow engine was somewhat unclear (Expert 3). It performed well in explaining the exclusive gateway and notations (Expert 3). Overall, more than 80% of the information was correct (Expert 5). However, some issues were also uncovered, such as the AI assistant confusing signals and messages, indicating the need for careful use of correct terminology where a message should be consistently referred to as a message and not a signal (Expert 4). In addition, the AI assistant initially omitted an activity when describing the minimum process and only correctly identified the "draft invoice" activity after repeated requests.

Table 9
Qualitative assessment of LLMs' ability to identify process model quality issues.

Process model	Evaluation criterion	*gemini-2.5-flash*	*gemini-2.5-pro*	*gpt-4.1-nano*	*o4-mini*
Dispatch of good	Issues Detected	1/1	1/1	0/1	1/1
	Explanation Quality	High	Medium	N/A	Medium
	Free from Hallucination?	✓	✓	×	×
Self-service restaurant	Issues Detected	2/3	3/3	0/3	2/3
	Explanation Quality	Medium	Medium	N/A	Medium
	Free from Hallucination?	✓	✓	✓	✓
Recourse	Issues Detected	1/3	2/3	0/3	1/3
	Explanation Quality	High	High	N/A	High
	Free from Hallucination?	✓	✓	✓	×
Credit scoring (Synchronous)	Issues Detected	1/1	1/1	0/1	1/1
	Explanation Quality	High	High	N/A	Medium
	Free from Hallucination?	✓	×	×	×
Credit scoring (Asynchronous)	Issues Detected	1/1	1/1	0/1	1/1
	Explanation Quality	High	High	N/A	High
	Free from Hallucination?	✓	✓	×	×

Table 10
Partners involved in the user-study.

Partner	Role	Company	Experience
Expert 1	Practice Lead Green BPM	Consultancy for Sustainable IT Applications	15–20 years
Expert 2	Green BPM Consultant	Consultancy for Sustainable IT Applications	0–5 years
Expert 3	Product and Innovation Manager	Software Developer for Process Automation Solutions	5–10 years
Expert 4	Co-Founder	Process Automation and Software Development	5–10 years
Expert 5	CoE Lead Process Mining	Automotive Manufacturer	0–5 years
Expert 6	BPM Expert	Semiconductor Manufacturer	5–10 years

Completeness (C2). In response to the question regarding the completeness of the answers, the feedback shows that the explanations are well-generalized and applicable to different elements of BPMN (Experts 1–6). The answers are effective for common processes such as the widely used ordering process and can be appropriately applied to similar scenarios. However, for unique or less typical processes, more scrutiny seems required to ensure the accuracy and applicability of the explanations (Expert 1). The AI assistant's ability to provide relevant, overarching explanations suggests that it has an appropriate level of generalizability for practical use in BPMN modeling.

Contextfullness (C3). For the question on contextual fulfillment, the AI assistant received mixed reviews with scores ranging from 3 to 5. While the AI assistant provided detailed and comprehensive answers that improved users' understanding of BPMN processes, concerns were raised that the AI assistant could mislead those unfamiliar with BPMN by relying too much on the AI assistant's explanations (Experts 1, 3). Users appreciated the insights relevant to process optimization, although Expert 5 perceived some responses as confusing, indicating the need for clearer and more precise contextual understanding. There would be further potential for improvement if the AI assistant could access a broader database for more specific analysis (Expert 5).

Interactiveness (C4). Regarding the question on interactivity, the performance of the tool was rated highly and predominantly scored between 4 and 5. It responded effectively to critical suggestions and responded appropriately to follow-up questions. Users appreciated that

Table 11
Criteria for evaluating the usability of explainable AI systems [36].

Criterion		Description
C1	Soundness	Ensures that the explanations accurately reflect the operations of the underlying model, emphasizing the truthfulness of the information presented.
C2	Completeness	Measures whether the explanation covers enough context and cases to be useful across various scenarios, not just the specific instance it was generated for.
C3	Contextfull-ness	Provides explanations with sufficient background to allow users to understand them in relation to their broader operational environment.
C4	Interactive-ness	Allows users to engage with the system dynamically, adjusting and querying the explanations to better suit their understanding or needs.
C5	Actionability	Ensures that the explanations guide users towards meaningful actions based on the insights provided.
C6	Chronology	Considers the timing of events or data points in explanations, highlighting the most recent or relevant information that impacts the outcome.
C7	Coherence	Focuses on the consistency of explanations with the user's prior knowledge and expectations, making them intuitively easier to accept.
C8	Novelty	Ensures that the information provided is insightful, revealing new information that can provoke thought or lead to unexpected insights.
C9	Complexity	Addresses the level of detail in the explanations, ensuring they are neither too simple to be trivial nor too complex to understand.
C10	Personaliza-tion	Tailors explanations to the background knowledge and specific needs of individual users, enhancing their relevance and accessibility.
C11	Parsimony	Emphasizes the brevity of explanations, ensuring they are concise and avoid overloading the user with unnecessary information.

the AI assistant could recall previous interactions (Expert 3). Although the AI assistant sometimes needed several attempts to give the correct answer (Expert 5), the responsiveness of the AI assistant to user queries was rated positively, which increases interactivity (Experts 2, 3, 6).

Actionability (C5). In addressing actionability, the AI assistant's performance varied, with ratings from 3 to 5. It was praised for providing valuable initial optimization insights that encouraged critical thinking and innovation (Expert 1, 2). Although some suggestions were too general (Expert 3), the AI assistant provided useful technical advice. Technical suggestions were beneficial; however, improvements in modeling and syntax could be better (Expert 6).

Chronology (C6). When asked about chronology, the AI assistant's ability to explain the chronological sequence of events in BPMN models was rated highly and generally received a rating of 5. It explained the sequence of events in different scenarios very effectively (Experts 1 – 6). Overall, the chronology was correctly followed in 100% of cases, although it should be noted that the scores were based on relatively simple models. It was pointed out that the AI assistant could reach its limits with more complex models (Experts 3, 5, 6).

Coherence (C7). In the assessment of coherence, the AI assistant scored well, with recommendations for clearer definitions of BPMN elements such as gateways and better clarity of process dependencies (Expert 1). Minor inaccuracies in the explanation of message flows and naming conventions were noted but did not significantly affect the overall coherence (Expert 3). These limitations emphasize the dependence of the LLM on its training data (Expert 4). Overall, the responses were logical and consistent and matched well with the available information.

Novelty (C8). In terms of novelty, the responses were generally not surprising (Experts 1, 5), but contained practical insights, e.g., on outsourcing activities (Expert 1). Although feedback on portfolio management was appreciated, the responses did not particularly impress those with extensive prior knowledge, suggesting that the novelty of the AI assistant's information may depend on the user's familiarity with the topic (Expert 5).

Complexity (C9). Evaluating complexity, the AI assistant mostly received high marks for the clarity and simplicity of its explanations (Experts 1, 3, 5). The answers were rated as very understandable and were generally in simple language. However, one criticism was that some answers were initially too long (Experts 4, 6), although the AI assistant was able to provide shorter answers when requested. It was also suggested that the lists within the responses should be formatted into paragraphs to improve readability and prevent the text from being overwhelming (Expert 6). Moreover, Expert 5 suggested that the answers should be revealed gradually, allowing users to begin reading while the response is still being generated.

Personalization (C10). In the evaluation of personalization, the AI assistant received mixed reviews. It received a high score for accurately addressing the most important use cases but was criticized for needing more diverse examples to improve its application (Expert 4). While the AI assistant showed good detail orientation, one expert noted that his knowledge exceeded the AI assistant's capabilities, indicating room for improvement in adapting to the experts' expectations (Expert 5).

Parsimony (C11). In the assessment of parsimony, the AI assistant was praised for its precision, particularly in the definition of technical terms and processes such as swim lanes, gateways, and data exchange, which are represented by dashed lines (Experts 1, 3). However, the answers were often felt to be too long. Suggestions for improvement included starting with shorter initial answers and only providing more detailed follow-up answers on request (Expert 4). While the explanations were generally precise, especially in terms of syntax, it was found that more open-ended questions led to less precise, detailed answers (Expert 6). Overall, while the AI assistant performed well in terms of comprehensiveness, there was a consensus that brevity could be improved.

Further feedback. Answers to the open-ended question in the third section were directed at the need for the AI assistant to provide its responses more quickly and to structure the responses in paragraphs for ease of reading (Expert 1). It was suggested that the AI assistant should clarify when its answers are merely suggestions that require further review and highlight the specific parts of BPMN that it is addressing. Users also wanted more interactive features, such as the ability to adapt the BPMN model directly through the tool to improve the user experience through more dynamic and human-like interactions (Expert 2). There was a desire for the tool to be able to enable the direct implementation of suggestions for improvement and interactive adjustments to the model (Expert 3).

8. Discussion and future directions

Our approach has demonstrated promising results in using LLMs to enhance the comprehension of process models, yet it also has limitations. This section outlines potential ideas for further research and development.

Aim for more concise outputs: The experts involved in the user study highlighted a tendency of *gpt-4* to produce long textual outputs. A more focused output is preferable to speed up the analytical reasoning. This could be addressed by changing the underlying LLM (for example, *Mixtral-8x22B-Instruct* produces significantly more concise outputs than other considered LLMs) or adopting new prompting strategies explicitly requesting the LLM to produce concise outputs. Also, it is possible to consider training/fine-tuning an LLM to align it with the desired output length.

Targeted prompt engineering: Despite the promising results of our experimentation, users noticed that LLMs sometimes fail to focus on the correct elements of the BPMN model given the inquiry. The size and complexity of the BPMN abstraction and the additional injected knowledge are overwhelming for the prompt attention mechanisms of current state-of-the-art LLMs. A possible solution is implementing targeted prompt engineering techniques, which involve focusing solely on the specific elements present in the uploaded process models or the user's query. For instance, a Retrieval-Augmented Generation (RAG) pipeline [37] could be implemented to store knowledge about different BPMN elements and retrieve the ones relevant for the uploaded model. Similarly, different types of few-shot examples can be stored, and the user's query can be analyzed to only retrieve relevant ones. However, the effectiveness of the retrieval is crucial and it should be thoroughly evaluated when integrated into the framework.

Providing process knowledge: LLMs are trained on a vast corpus of data, which allows them to be a general-purpose conversational interface. They also show good domain knowledge in popular benchmarks. However, LLMs may not know in detail how a given process has been implemented in a given setting. Therefore, a user inquiry might be enriched with relevant contextual information about the process. This can be implemented using a RAG pipeline, or by fine-tuning the LLM on specific process knowledge.

Enhanced interactivity: Our framework utilizes LLMs to produce a textual answer to the inquiry of the user. However, analysts would benefit from a feedback mechanism in which the parts of the BPMN model mentioned in the answer are visually highlighted. This would require the specification of a clear output structure to the LLM, in which the textual answer and the configuration of the highlighting are the components. Furthermore, experts from the user study suggested empowering users to implement suggestions for improvement and make interactive adjustments to the BPMN model directly within AIPA. Such features could foster a more responsive and engaging user experience.

What if one does not know what to ask? LLMs prove useful in answering user inquiries over a BPMN model. However, also thinking about which inquiries should be asked in order to achieve a complete understanding of the process requires significant expertise by the user.

This could be addressed within our framework by integrating a mechanism for automated hypothesis generation, which can also be based on LLMs. This mechanism could help in identifying a list of relevant questions given the process model and the conversational history.

What if the process is not modeled in BPMN? The proposed framework is useful for comprehending BPMN models. However, many organizational processes are expressed in textual documents. To use our framework on such processes, *process modeling* could be applied to get a BPMN model from the textual representation. Integrating process modeling and comprehension techniques would allow process analysts to fully understand the entire set of processes of an organization, even the ones not modeled in BPMN.

Integration with process mining: In future work, we aim to integrate our LLM-based process model comprehension framework with process mining methodologies. In particular, we plan to employ LLMs to interpret and analyze process models that are automatically discovered from event data and annotated with performance metrics or compliance deviations. This approach will bridge the gap between theoretical models and actual process executions, aligning our developments with real-world process dynamics.

Exploring further LLM applications in BPM: As we continue to expand the capabilities of LLMs within the BPM field, their potential extends beyond improving process model comprehension. For example, there is a potential to apply LLMs for process enhancement. Future research could investigate how LLMs can not only analyze but also recommend optimizations or variations in process flows.

9. Conclusion

In this paper, we introduced a novel framework that utilizes the advanced natural language processing capabilities of Large Language Models (LLMs) to enhance the comprehension of complex process models. We transform intricate Business Process Model and Notation (BPMN) diagrams into various abstraction formats suitable for LLM interpretation, and we explore various prompting strategies to optimize LLM performance. Furthermore, we presented AIPA (AI-Powered Process Analyst), a tool developed to integrate our framework with various state-of-the-art LLMs. AIPA supports dynamic interactions with process models, enabling users to query BPMN models and receive explanations seamlessly.

We extensively evaluated our framework with different BPMN abstractions and prompting strategies. Our evaluation results indicate that the right combination of model abstraction and prompting strategies significantly improves the model's comprehensibility without compromising detail or accuracy. A qualitative analysis also demonstrated the framework's capability for model quality assessment, showing that LLMs can identify critical structural flaws, though their inconsistent performance underscores the need for expert validation. Moreover, we conducted a user study to assess the ease of comprehension, accuracy of interpretation, and user satisfaction when interacting with BPMN models through AIPA. Results demonstrate a marked improvement in understanding complex models with our LLM-based framework, confirming its effectiveness and user-friendliness. This paper not only underscores the potential of LLMs to make BPMN models more accessible but also provides empirical evidence, highlighting the transformative potential of leveraging AI technologies for BPM and process mining tasks.

CRediT authorship contribution statement

Humam Kourani: Writing – review & editing, Writing – original draft, Software, Methodology. **Alessandro Berti:** Writing – review & editing, Writing – original draft, Software, Methodology. **Jasmin Hennrich:** Writing – review & editing, Writing – original draft, Methodology. **Wolfgang Kratsch:** Writing – review & editing, Methodology. **Robin Weidlich:** Writing – review & editing, Writing – original draft, Methodology. **Chiao-Yun Li:** Writing – original draft, Conceptualization. **Ahmad Arslan:** Software. **Wil M.P. van der Aalst:** Writing – review & editing, Conceptualization. **Daniel Schuster:** Writing – review & editing, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

We have shared a link to the data and code in the paper.

References

[1] M. Dumas, M.L. Rosa, J. Mendling, H.A. Reijers, Fundamentals of Business Process Management, Second Edition, Springer, 2018.
[2] M. von Rosing, S. White, F. Cummins, H. de Man, Business process model and notation - BPMN, in: The Complete Business Process Handbook: Body of Knowledge from Process Modeling To BPM, vol. I, Morgan Kaufmann/Elsevier, 2015, pp. 429–453.
[3] M. zur Muehlen, J. Recker, How much language is enough? Theoretical and practical use of the business process modeling notation, in: CAiSE 2008, in: LNCS, vol. 5074, Springer, 2008, pp. 465–479.
[4] J. Recker, Opportunities and constraints: the current struggle with BPMN, Bus. Process. Manag. J. 16 (1) (2010) 181–201.
[5] P. Bera, Does cognitive overload matter in understanding BPMN models? J. Comput. Inf. Syst. 52 (4) (2012) 59–69.
[6] J. Huang, K.C. Chang, Towards reasoning in large language models: A survey, in: ACL 2023, Association for Computational Linguistics, 2023, pp. 1049–1065.
[7] C. Zhou, D. Zhang, D. Chen, C. Liu, Business Process Complexity Measurement: A Systematic Literature Review, IEEE Access 11 (2023) 47940–47955.
[8] A.A. Andaloussi, D. Lübke, B. Weber, Conducting eye-tracking studies on large and interactive process models using EyeMind, SoftwareX 24 (2023) 101564.
[9] M. Winter, C. Bredemeyer, M. Reichert, H. Neumann, R. Pryss, A comparative cross-sectional study on process model comprehension driven by eye tracking and electrodermal activity, 2023.
[10] A. Polyvyanyy, Business process querying, in: Encyclopedia of Big Data Technologies, Springer, 2019.
[11] P. Delfmann, D.M. Riehle, S. Höhenberger, C. Corea, C. Drodt, The diagramed model query language 2.0: Design, implementation, and evaluation, in: Process Querying Methods, Springer, 2022, pp. 115–148.
[12] H. Störrle, V. Acretoaie, VM*: A family of visual model manipulation languages, in: Process Querying Methods, Springer, 2022, pp. 149–179.
[13] C. Di Francescomarino, P. Tonella, The BPMN visual query language and process querying framework, in: Process Querying Methods, Springer, 2022, pp. 181–218.
[14] K. Kammerer, R. Pryss, M. Reichert, Retrieving, abstracting, and changing business process models with PQL, in: Process Querying Methods, Springer, 2022, pp. 219–254.
[15] A. Polyvyanyy, Process query language, in: Process Querying Methods, Springer, 2022, pp. 313–341.
[16] M. Proietti, F. Taglino, F. Smith, QuBPAL: Querying business process knowledge, in: Process Querying Methods, Springer, 2022, pp. 255–284.
[17] M.L. Rosa, H.A. Reijers, W.M.P. van der Aalst, R.M. Dijkman, J. Mendling, M. Dumas, L. García-Bañuelos, APROMORE: an advanced process model repository, Expert Syst. Appl. 38 (6) (2011) 7029–7040.
[18] A. Ligeza, T. Potempa, AI approach to formal analysis of BPMN models: Towards a logical model for BPMN diagrams, in: ABICT 2012, in: AISC, vol. 257, Springer, 2012, pp. 69–88.
[19] A. Casciani, M.L. Bernardi, M. Cimitile, A. Marrella, Conversational systems for AI-augmented business process management, in: RCIS 2024, in: LNBIP, vol. 513, Springer, 2024, pp. 183–200.
[20] A. Berti, D. Schuster, W.M.P. van der Aalst, Abstractions, scenarios, and prompt definitions for process mining with LLMs: A case study, in: BPM 2023 Workshops, in: LNBIP, vol. 492, Springer, 2023, pp. 427–439.
[21] A. Berti, M.S. Qafari, Leveraging large language models (LLMs) for process mining, Technical Report), 2023, CoRR abs/2307.12701.
[22] M.L. Bernardi, A. Casciani, M. Cimitile, A. Marrella, Conversing with business process-aware large language models: the BPLLM framework, 2024.
[23] D. Fahland, F. Fournier, L. Limonad, I. Skarbovsky, A.J. Swevels, How well can large language models explain business processes?, 2024, arXiv preprint arXiv:2401.12846.
[24] R. Tang, D. Kong, L. Huang, H. Xue, Large language models can be lazy learners: Analyze shortcuts in in-context learning, in: ACL 2023, Association for Computational Linguistics, 2023, pp. 4645–4657.
[25] A. Kong, S. Zhao, H. Chen, Q. Li, Y. Qin, R. Sun, X. Zhou, Better zero-shot reasoning with role-play prompting, 2023, CoRR abs/2308.07702.

[26] B. Xu, A. Yang, J. Lin, Q. Wang, C. Zhou, Y. Zhang, Z. Mao, ExpertPrompting: Instructing large language models to be distinguished experts, 2023, CoRR abs/2305.14688.
[27] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E.H. Chi, Q.V. Le, D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, in: NeurIPS 2022, 2022.
[28] N. Kazi, I. Kahanda, Enhancing transfer learning of LLMs through fine- tuning on task - related corpora for automated short-answer grading, in: ICMLA 2023, IEEE, 2023, pp. 1687–1691.
[29] A. Martino, M. Iannelli, C. Truong, Knowledge injection to counter large language model (LLM) hallucination, in: ESWC 2023 Satellite Events, in: LNCS, vol. 13998, Springer, 2023, pp. 182–185.
[30] T.B. Brown, B. Mann, N. Ryder, M. Subbiah, et al., Language models are few-shot learners, in: NeurIPS 2020, 2020.
[31] D. Miyake, A. Iohara, Y. Saito, T. Tanaka, Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models, 2023, CoRR abs/2305.16807.
[32] J. Munoz-Gama, R. de la Fuente, M. Sepúlveda, R. Fuentes, Conformance checking challenge 2019, 2019.
[33] A. Berti, H. Kourani, H. Häfke, C. Li, D. Schuster, Evaluating large language models in process mining: Capabilities, benchmarks, and evaluation strategies, in: BPMDS 2024, 2024, pp. 13–21.
[34] Z. Ji, T. Yu, Y. Xu, N. Lee, E. Ishii, P. Fung, Towards mitigating hallucination in large language models via self-reflection, 2023, CoRR abs/2310.06271.
[35] U. Schultze, M. Avital, Designing interviews to generate rich data for information systems research, Inf. Organ. 21 (1) (2011) 1–16.
[36] K. Sokol, P.A. Flach, Explainability fact sheets: a framework for systematic assessment of explainable approaches, in: FAT* '20, ACM, 2020, pp. 56–67.
[37] P.S.H.L. et al., Retrieval-augmented generation for knowledge-intensive NLP tasks, in: NeurIPS 2020, December 6-12, 2020, Virtual, 2020.

Humam Kourani is a Research Associate in the Data Science and Artificial Intelligence Department at the Fraunhofer Institute for Applied Information Technology (FIT). He leads research and software development projects and serves as an examiner at the Fraunhofer Personnel Certification Authority. Humam is also pursuing his Ph.D. in the Process and Data Science (PADS) group at RWTH Aachen University. His research primarily focuses on business process modeling, process discovery, and the integration of large language models into process mining.

Alessandro Berti is a Ph.D. student at RWTH Aachen University, affiliated with the Process and Data Science (PADS) group. His doctoral thesis focuses on object-centric process mining. He plays a role as the main developer of pm4py, a leading Python library for process mining. Alessandro has made significant contributions to the integration of large language models within the pm4py framework. His work includes both development and research, with some publications that bridge the gap between large language models and the field of process mining.

Jasmin Hennrich is a research assistant at FIM Research Center and Branch Business & Information Systems Engineering of the Fraunhofer FIT and a Ph.D. candidate at the University of Bayreuth since June 2020. Her research interests focus on the adoption of artificial intelligence applications in practice. Specifically, she investigates factors influencing the adoption and measures enabling the adoption of artificial intelligence applications, primarily in the healthcare sector. Her research has been published in journals such as Journal of Medical Internet Research and BMC Health Services, as well as presented at conferences like the International Conference on Information Systems and the European Conference on Information Systems.

Wolfgang Kratsch studied Computer Science and Information Management at the University of Augsburg. Wolfgang gained his Ph.D. at the University of Bayreuth in December 2020. In 2020, Wolfgang also co-founded the Fraunhofer spinoff credium. He also co-founded the Fraunhofer Center for Process Intelligence (CPI). Since 2022, Wolfgang holds the professorship for Applied AI in Digital Value Creation at the Technical University of Applied Sciences Augsburg and is additionally affiliated as Director of the FIM Research Center for Information Management. He works in leading position with the Branch Business & Information Systems Engineering of the Fraunhofer FIT. Most of Wolfgang's work centers around process mining and process analytics.

Robin Weidlich studied Industrial Engineering and Management (M.Sc.) at the University of Bayreuth. During his studies, he gained practical experience in internships in different fields of the automotive industry. Robin has been working as a research associate and doctoral candidate at the FIM Research Center and Branch Business & Information Systems Engineering of the Fraunhofer FIT since January 2022. He is primarily interested in the digitalization of business processes and data-driven business process management empowered by emerging technologies, such as artificial intelligence and the Internet of Things as well as the application of such technologies in the healthcare sector. His research has been published in journals such as Business Process Management Journal, Communications of the Association for Information Systems and European Journal of Information Systems.

Chiao-Yun Li is a Ph.D. candidate at the Chair of Process and Data Science at RWTH Aachen University. Her research interests mainly focus on event abstraction on partially ordered event data. She also extends her research in applying process mining in practice. Specifically, she has led various industrial projects in the field of process mining. Her research is published in various process mining conferences, including International Conference on Business Process Management (BPM) and International Conference on Process Mining (ICPM).

Ahmad Arslan is a software engineer currently pursuing a master's degree in data science, specializing in process mining, at RWTH Aachen University. He brings extensive experience as a full-stack developer, having worked in diverse fields, including workflow management in healthcare and finance, for both startups and multinational enterprises.

Wil van der Aalst is a full professor at RWTH Aachen University, leading the Process and Data Science (PADS) group. He is also the Chief Scientist at Celonis, part-time affiliated with the Fraunhofer FIT. He also has unpaid professorship positions at Queensland University of Technology (since 2003) and the Technische Universiteit Eindhoven (TU/e). Currently, he is also deputy CEO of the Internet of Production (IoP) Cluster of Excellence and co-director of the RWTH Center for Artificial Intelligence. His research interests include process mining, Petri nets, business process management, workflow management, process modeling, and process analysis. Wil van der Aalst has published over 275 journal papers, 35 books (as author or editor), 675 refereed conference/workshop publications, and 85 book chapters.

Daniel Schuster is a senior scientist affiliated with the Fraunhofer Institute for Applied Information Technology FIT, where he leads the process mining research group. He is also affiliated with the chair of Process and Data Science at RWTH Aachen University. His research interests include hybrid intelligence in process mining, particularly interactive and incremental process discovery.