Conversationally Actionable Process Model Creation

Nataliia Klievtsova[1], Timotheus Kampik[2], Juergen Mangler[1], and Stefanie Rinderle-Ma[1]

[1] Technical University of Munich, Germany; TUM School of Computation, Information and Technology
{nataliia.klievtsova,juergen.mangler,stefanie.rinderle-ma}@tum.de
[2] SAP Signavio, Berlin, Germany, timotheus.kampik@sap.com

Abstract. With the recent success of large language models, the idea of AI-augmented Business Process Management systems is becoming more feasible. One of their essential characteristics is the ability to be conversationally actionable, allowing humans to interact with the system effectively. However, most current research focuses on single-prompt execution and evaluation of results, rather than on continuous interaction between the user and the system. In this work, we aim to explore the feasibility of using chatbots to empower domain experts in the creation and redesign of process models in an effective and iterative way. In particular, we experiment with the prompt design for a selection of redesign tasks on a collection of process models from literature. The most effective prompt is then selected for the conducted user study with domain experts and process modelers in order to assess the support provided by the chatbot in conversationally creating and redesigning a manufacturing process model. The results from the prompt design experiment and the user study are promising w.r.t. correctness of the models and user satisfaction.

Keywords: Process Discovery · Process Models · Large Language Models · Process Improvement · Conversations

1 Introduction

Business process modeling is an approach to describe how businesses execute their operations [10] by using graphical constructs to describe and implement the business logic. The utilization of a standardized notation such as Business Process Model and Notation (BPMN 2.0[3]) typically improves operational efficiency, significantly minimizes errors, and enhances communication and collaboration. One of the primary challenges is the extensive training and skill development required for best-practice utilization of BPMN by various stakeholders within an organization, such as domain experts and process designers/modelers. The successful creation of best-practice models [33] can be facilitated either by extensive

[3] www.omg.org/spec/BPMN/2.0

collaboration between domain experts and modelers, or by investing in training programs for domain experts, so that they can do the modeling themselves.

While collaborations help to avoid the implementation of special training programs and ensure that BPMN models are well designed [33], they can also lead to a "dilemma between process modeler and domain expert" as there is no or only limited knowledge overlap between them, i.e., there exists a communication gap. The process modeler lacks specific domain knowledge, while the domain expert may have only limited knowledge of process model notations [28]. The constant need to transfer the domain knowledge to process modelers is especially burdensome for organizations continuously undergoing adaptations caused by internal or external changes, i.e., when business processes need to be designed or redesigned to improve their day-to-day execution performance [7]. Hence, it is crucial to find a simple and effective way to generate, manipulate, and evaluate process models, minimizing the communication effort of domain experts.

Conversational process modeling (CPM) [23] aims to maximize the involvement of domain experts in the creation of process models and hence to minimize the communication effort between domain experts and process modelers [27]. Specifically, CPM refers to the iterative process of creating process models based on process descriptions and conversations between domain experts and chatbots, until the created models reach a certain quality level and become sufficiently mature to fulfill their purpose. This paper advances our previous work on CPM [23,24] by providing an in-depth evaluation of whether domain experts can design and redesign a process model in a conversationally actionable manner, i.e., in interaction with a chatbot instead of a process modeler.

In Sect. 2, we explore the process of process model creation from the perspective of a domain expert by employing Large Language Models (LLMs) as a conversational tool that substitutes the process modeler. Section 3 demonstrates the capabilities of LLMs, such as GPT-4, for model redesign and refinement, in connection with the textual representation of graphical notation of the process model using the JavaScript-based visualization library Mermaid.js. In particular, we experiment with different redesign tasks based on change patterns from literature [39] for finding the most effective prompt design. At this, effectiveness is assessed based on the syntactic and semantic correctness of the resulting process models. Moreover, a user study is conducted to assess the quality of the LLM-redesigned models regarding user satisfaction, model completeness and correctness, layouting and the quality of the selected graphical representation (see Sect. 4). Section 5 discusses related work and Sect. 6 concludes the paper.

2 Conversationally Actionable Process Model Creation

One future direction in business process management is the development of AI-augmented process-aware information systems, i.e., systems that act in an autonomous, adaptive, explainable, and *conversationally actionable* way [9,12]. In the following, we examine the aspect of how to make process models creation conversationally actionable, i.e., allowing domain experts to create and redesign process models interactively in a conversation with the system via a chatbot. To this end, we start with an analysis of the process of process model creation as currently applied and realized as interaction between domain expert and process modeler. We show how this process can be transformed into the conversationally actionable process model creation (CAPMC) approach, based on interaction between domain expert and chatbot.

2.1 Interaction in Process Model Creation

A common issue in process model creation is the complexity of the modeling notation, making it challenging for domain experts, as they possess the knowledge about their application domains and typically lack modeling knowledge [36]. The latter hinders the creation of correct models as well as the analysis of models for errors. This leads to substantial efforts being spent on training domain experts in process modeling, diverting resources away from solving business problems, such as simplifying, enhancing, and optimizing these processes [36].

One typically used remedy strategy is to team up the domain expert with a process modeler who has extensive knowledge on how to create correct process models and typically lacks knowledge about the application domain. The process modeler then creates the process model based on her interpretation of the knowledge provided by the domain experts [22]. The cooperation between a process modeler and a domain expert can be established in multiple ways distinguished by diverse methods of information gathering about a process. According to [13] there are three types of discovery methods, i.e., evidence-based discovery, interview-based discovery, and workshop-based discovery.

In all cases, a process modeler can either play (1) an active role (i.e., direct communication) performing one-on-one interviews with domain experts, or (2) facilitate a series of modeling sessions in which several domain experts come together to negotiate a common view of the global process model [22]. Additionally the process modeler can act (3) as passive observer, studying the evidence to get familiar with certain parts of a process and its environment, and to formulate hypotheses [13]. Direct process modeling interaction (1) is depicted in Fig. 1. Here, the interpretation of the process description and the changes provided by the domain expert are interpreted by the process modeler which might lead to validity errors due to misunderstandings and the modeler being agnostic of the application domain.

2.2 Chatbot vs Process Modeler: Does the Difference Matter?

Process model creation is generally most efficiently conducted when domain knowledge can be accessed immediately by individuals directly involved in the process. This helps reduce the risk that the modeling expert becomes a bottleneck for capturing process knowledge [22].

Natural text-based language is one of the preferred process representations among domain experts, primarily due to lack in experience and knowledge in process modeling [2,26]. Therefore, it is necessary to develop a framework where

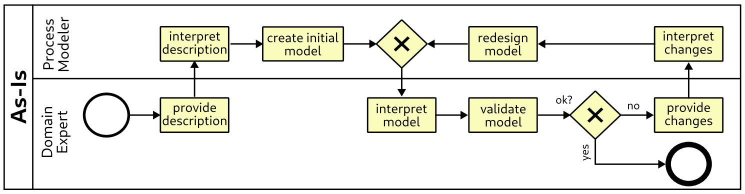


Fig. 1. Interaction between Domain Expert and Process Modeler (1)

domain experts can create and iteratively refine a model using natural language, in a human-like conversational manner and without the involvement of a (human) process modeler. This is realized in the Conversational Process Modeling (CPM) framework introduced in our previous work [23]; the core CPM process is depicted in Fig. 2[4]. Task `refine description/model` is a complex task and the underlying sub process defines the iterative interaction between text to model (T2M) transformation performed by a chatbot and model interpretation and its redesign via process description adjustment performed by a domain expert. A process model is created based on a process description and it is reviewed and adjusted multiple times before it is used further. We refer to this iterative and continuous interplay between model generation and its interpretation as *Conversationally Actionable Process Model Creation*, defined in Concept 1.

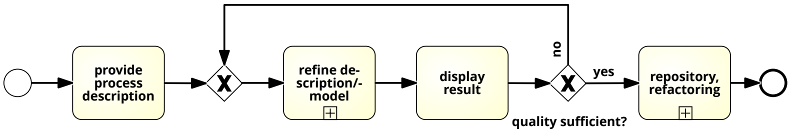


Fig. 2. Conversational Process Modeling, abstracted from [23]

Concept 1 (Conversationally Actionable Process Model Creation) *Conversationally Actionable Process Model Creation (CAPMC) refers to the continuous translation and interpretation of modeling artifacts utilized during the real-time interaction between a chatbot and different stakeholders involved in process design. With real-time we refer here to the immediate, live exchange of messages as in a human-like communication. Under these circumstances, stakeholders interact with the chatbot using domain-specific natural language (**conversationally**), and the chatbot reacts to their requests (**actionable**), translating them into process models (**process model creation**).*

CAPMC, as realized by interactions between domain expert and chatbot is depicted in Fig. 3. Note that according to Concept 1, further stakeholders such

[4] We abstract from the tasks referring to storing and refactoring of process models which can be targeted in future work.

as the process modeler can be involved in CAPMC, as well. In this work, we aim at contrasting traditional process model creation with process modeler as depicted in Fig. 1 with CAPMC without process modeler as depicted in Fig. 3. In the latter case, the domain expert provides her knowledge in the form of a process description. Based on the process description, the chatbot creates an initial process model. The model is then interpreted and validated by the domain expert. If the validation fails, changes are provided by the domain expert to the chatbot. The model is then again interpreted and validated by the domain expert. This change and validate cycle is repeated until the model reaches a certain validation quality. Note that for soundness checks of the model, at each point, the process modeler and/or automatic soundness checks can be applied.

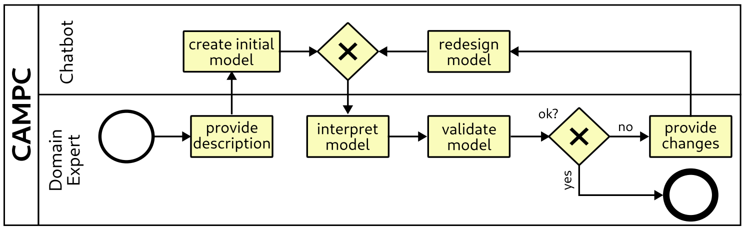


Fig. 3. CAMPC between Domain Expert and Chatbot

We compare CAPMC with chatbot (cf. Fig. 3) to traditional process model creation with process modeler (cf. Fig. 1). Starting with the effort of the domain expert, she is supposed to conduct the same tasks for both approaches, i.e., `interpret model`, `validate model`, and `interpret model`. Moreover, when modeling without a human process modeler, the domain expert can face several issues such as lack of specialized skills, technical problems handling modeling tools, modeling errors, and as follow intensive time commitment. Furthermore, complexity management and maintenance of the model, communication issues with other participants who rely on the model, as well as the limitations when using BPMN modeling tools (e.g., interoperability, user interface complexity, insufficient functionality, lack of customization, and inadequate technical documentation, etc.) can also lead to problems.

However, the utilization of a chatbot offers advantages at multiple levels that help overcome the limitations mentioned above. First, while a domain expert communicates with the chatbot instead of a process modeler, there tasks `interpret description` and `interpret changes` (cf. Fig. 1) become obsolete, and consequently, additional documentation or conversations to prevent substantial errors and failures are not required [22]. Second, one of the primary challenges, i.e., finding a common language between modeling language and domain-specific natural language [34], is effectively overcome. Third, a direct involvement in process model design encourages a sense of psychological ownership, which has been demonstrated to positively impact not only affective commitment but also the quality of the model [16].

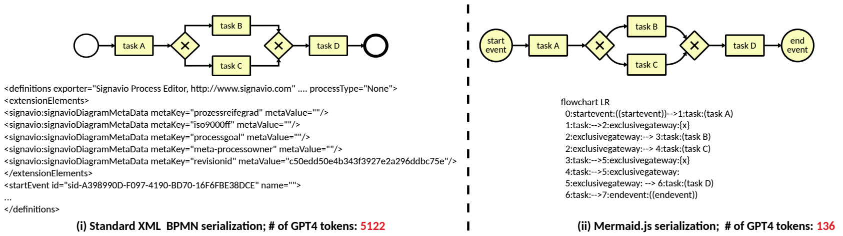
CAPMC with chatbot is expected to only create models that adhere to the syntactic correctness requirements of the used process modeling notation. This prevents the domain expert from introducing syntactic errors and allows her to focus solely on the semantic aspects of process modeling. The domain expert can immediately inspect and interpret a created model. As soon as an error or an inconsistency is detected, the domain expert can promptly refine the model. The advantage of this form of interaction is that the domain expert can interact with the chatbot as frequently as desired, without fear of negative judgment from the chatbot [19]. Since the models are created based on text provided by the domain expert, any occurring errors or mistakes can be considered a part of the learning process, contributing to the development of modeling skills and process thinking. Furthermore, there is evidence that a manual analysis of the created models and the documented interactions between domain expert and chatbot can assist novice analysts in the creation and optimization of these models [38].

3 Model Redesign for LLMs

As depicted in Fig. 3, in CAPMC, the domain expert is interacting with the chatbot where the chatbot serves as interface to an underlying LLM performing tasks `create inital model` and `redesign model`. We have provided LLM-based methods for creating process models from text, i.e., process descriptions, in our previous work [23,24]. In the following, we focus on task `redesign model` based on LLMs, i.e., the selection of suitable graphical representations and prompt engineering.

Selection of Graphical Representation: The context window of an LLM refers to the maximum number of tokens that can be put into the model at a time. This number is limited because LLMs are trained with a fixed length of training sequences [20]. Most regular large language models have a context window limit between 1,000 and 8,000 tokens. Due to its complexity, a simple BPMN model with 4 tasks in XML representation might exceed 4,000 to 8,000 tokens [1]. However, most of the information provided in this representation is not specific to the process content, i.e., specifies, for example, the layouting or the boilerplate overhead on diagram and element levels. Therefore, a simplified abstract representation of the BPMN model is required to enable process model generation and direct visualization of the output from an LLM in a user-friendly manner. We select Mermaid.js (MER) as our representation format because it is widely used, well-documented, and yields consistent results during model generation (see Fig. 4). Moreover, Mermaid.js performs well in model creation from text as shown in our previous work in [24].

Prompt Engineering: The construction of a prompt to guide LLMs in performing a required task in the most efficient way, known as prompt engineering, is typically used to avoid fine-tuning [29]. This approach makes general-purpose LLMs more task-specific [3]. For initial model generation, we utilize the prompt



```
<definitions exporter="Signavio Process Editor, http://www.signavio.com" .... processType="None">
<extensionElements>
<signavio:signavioDiagramMetaData metaKey="prozessreifegrad" metaValue=""/>
<signavio:signavioDiagramMetaData metaKey="iso9000ff" metaValue=""/>
<signavio:signavioDiagramMetaData metaKey="processgoal" metaValue=""/>
<signavio:signavioDiagramMetaData metaKey="meta-processowner" metaValue=""/>
<signavio:signavioDiagramMetaData metaKey="revisionid" metaValue="c50edd50e4b343f3927e2a296ddbc75e"/>
</extensionElements>
<startEvent id="sid-A398990D-F097-4190-BD70-16F6FBE38DCE" name="">
...
</definitions>
```

(i) Standard XML BPMN serialization; # of GPT4 tokens: 5122

```
flowchart LR
  0:startevent:((startevent))-->1:task:(task A)
  1:task:-->2:exclusivegateway:{x}
  2:exclusivegateway:--> 3:task:(task B)
  2:exclusivegateway:--> 4:task:(task C)
  3:task:-->5:exclusivegateway:{x}
  4:task:-->5:exclusivegateway:
  5:exclusivegateway: --> 6:task:(task D)
  6:task:-->7:endevent:((endevent))
```

(ii) Mermaid.js serialization; # of GPT4 tokens: 136

Fig. 4. Business Process represented via BPMN2.0 and Mermaid.js

provided in [24], which yields good results in terms of completeness and correctness of the created process model. For model redesign, we adopt the structure of this prompt and create rules defining the desired output format of the redesigned model. Figure 6 depicts the final structure of the prompt for process model redesign. The prompt consists of three parts: [1] provided changes, [2] additional information, and [3] the actual task that should be executed by the LLM. Central for the redesign are the changes provided by the domain expert based on her assessment of the current process model. Change of process models can be represented in different ways, e.g., based on change patterns [39]. CAPMC allows changes to be stated in natural language. Assume, for example, a current process model where tasks A, B, and C form a sequence, and the domain expert states the following change: "task A should be executed in parallel with tasks B and C" (see Fig. 5).

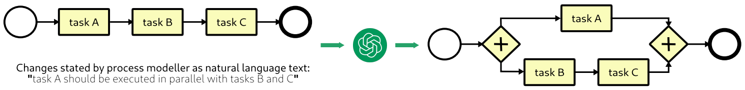

task A
task B
task C
Changes stated by process modeller as natural language text:
"task A should be executed in parallel with tasks B and C"
task A
task B
task C

Fig. 5. Model Redesign utilizing Chatbot

Based on the combination of the elements described in each part in Fig. 6, we developed five prompts [5]. The changes [1], specified by the user for model redesign, and the actual task to be performed by the LLM [3] are included in each prompt. Optional information [2] is inserted in various combinations to explore its influence on the quality of the LLM-generated output.

Prompt Selection: To select the most suitable prompt for model redesign, we define four redesign tasks in natural language as representatives for common change and adaptation patterns (AP), i.e., insertion and deletion of tasks (AP1 and AP2 from [39]) where insertion is further varied into conditional (AP10) and parallel (AP9) insertion. During prompt selection, we focus on these change patterns as the simplest building blocks that enable users to create and redesign process models [40] and because most modeling environments rely on these fun-

[5] https://github.com/com-pot-93/campc/tree/main/prompt_engineering

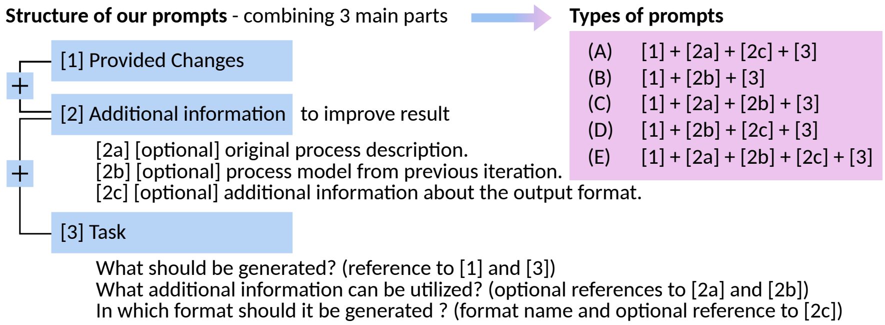

Structure of our prompts - combining 3 main parts
Types of prompts
[1] Provided Changes
+
[2] Additional information to improve result
[2a] [optional] original process description.
[2b] [optional] process model from previous iteration.
[2c] [optional] additional information about the output format.
+
[3] Task
What should be generated? (reference to [1] and [3])
What additional information can be utilized? (optional references to [2a] and [2b])
In which format should it be generated ? (format name and optional reference to [2c])
(A) [1] + [2a] + [2c] + [3]
(B) [1] + [2b] + [3]
(C) [1] + [2a] + [2b] + [3]
(D) [1] + [2b] + [2c] + [3]
(E) [1] + [2a] + [2b] + [2c] + [3]

Fig. 6. Prompt Structure and Types of Prompts

damental operations. In future work, we plan to incorporate more change patterns into.

The prompt selection is then performed in two rounds. First, we apply the synthetic redesign tasks to 7 different process models, ensuring that the redesign tasks are similar for each model regardless of the model complexity and domain (Round 1). Then, we focus on a single process model, adjust the redesign tasks according to its specific description, and apply the best prompts identified from the first round (Round 2). Applying the synthetic redesign tasks to different process models from different domains (Round 1) aims at selecting the most effective prompts for model redesign across different domains providing a broad evaluation. Round 2 is supposed to ensure performance for process model redesign within a more specific context. Moreover, the redesign tasks for Rounds 1 and 2 are designed to be straightforward and uniform serving to create a common ground despite potential differences in phrasing during communication with real domain experts.

Round 1: The following 4 synthetic redesign tasks are utilized for Round 1:

(a) add a task 'dummy task 1' after the second task;
(b) delete the third task in the model;
(c) add an alternative branch with the task 'dummy task 2' for the second task;
(d) add a task 'dummy task 3' parallel to the first task;

We apply these changes to multiple process descriptions to evaluate how effectively these changes are adapted by an LLM[6]. We utilize GPT-4, as currently it is considered one of the leading LLMs. As process descriptions, we utilize examples from the PET dataset [5] describing processes from multiple domains. The processes comprise between 3 and 11 tasks, at least 2 events, exclusive and parallel gateways, and involve 2 or more participants. Figure 7 depicts an "original" process model, e.g., a created process model in a certain iteration of the CAPCM interaction between domain expert and chatbot. Assume that the domain expert assesses the model and notes that a task D is missing after the

[6] https://lmsys.org/blog/2023-06-22-leaderboard/

first task and specifies this as the change shown in Fig. 7 (middle part at the top), using change template example (a) with taks D as dummy task.

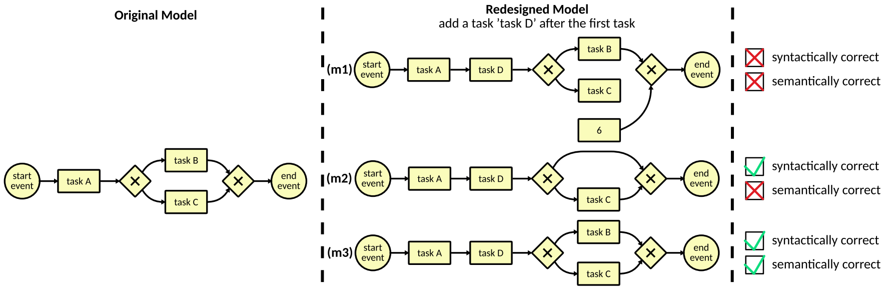


Fig. 7. Syntactic and Semantic Correctness of a Model

A redesign task is then assessed w.r.t. its *syntactic* and *semantic* correctness where it is assumed that the original model is a valid mermaid.js model according to the Mermaid.js specification[7]. We assume that a redesign task can be applied to a given model, i.e., the preconditions as stated in [35] are fulfilled, e.g., one can only delete a task that exists in the model. A redesign task is considered *syntactically correct* if, after its execution, the newly created model is again a valid mermaid.js and adheres to the predefined output format as specified in the prompt. Moreover, a redesign task is considered *semantically correct* if it fulfills the post-conditions as stated in [35], e.g., tasks are added at the intended position. Moreover, the LLM should not hallucinate w.r.t. change, i.e., create effects that are not specified in the redesign task such as inserting arbitrary tasks. In Fig. 7, created model (m1) is syntactically not correct due to node `6` having no incoming edge and semantically not correct as `6` was inserted, but not specified in the redesign task.

We start with checking syntactic correctness of the created model, i.e., verify if the model is valid and check whether its textual notation adheres to the predefined output format[8]. If syntactic correctness is satisfied, we evaluate whether the changes specified in the redesign task were performed correctly by the LLM. We check if the desired element was added, if it was added in the correct position, and if all accompanying attributes were added correctly (i.e., if gateways were also added by parallel tasks or decisions). Additionally, we check if other elements in the model remain unchanged.

During evaluation, prompt A was excluded since its design utilizes only a textual process description and a redesign task as input. This causes complications because, before submitting the output model to the LLM for the next iteration, we need to convert it into a textual description to maintain the modifications made.

[7] https://mermaid.js.org/syntax/flowchart.html

[8] https://github.com/com-pot-93/campc/blob/main/prompt_engineering/prompts.txt

Table 1 summarizes the evaluation results. The best results are achieved with prompt B, where all created models are syntactically correct and the application of redesign tasks (a) and (b) achieve also 6 out of 7 semantically correct models. Redesign tasks (c) and (d) result in a low number of semantically correct process models. Prompt C shows the lowest performance, creating a limited number of syntactically correct models in comparison to the other prompts. The greater the number of changes made to the model, the more the current model deviates from the original process description. We suggest that inconsistencies between the original process description and the redesign task result in dubious outcomes. Prompt E also results in syntactically correct models and is obviously less prepared to deal with deleting tasks as Prompt B, though it has the largest amount of additional information. Prompt D creates a limited number of semantically correct models (see Tab. 1).

Table 1. Prompt Selection Assessment: Round 1 (7 Models)

	Prompt B		Prompt C		Prompt D		Prompt E	
	syntactic	semantic	syntactic	semantic	syntactic	semantic	syntactic	semantic
(a)	7	6	7	6	7	6	7	6
(b)	7	6	7	4	7	5	7	4
(c)	7	2	5	3	6	0	7	3
(d)	7	0	4	0	6	0	7	0
sum	28	14	23	13	26	11	28	13

Generally, we can see that easier redesign tasks (a) and (b) show a better performance than the more sophisticated ones for conditional (c) and parallel insert (d). Most of the semantic errors occur due to a misinterpretation of the redesign task (e.g., the task is inserted in the wrong position, or the wrong task is deleted, or additional task is deleted).

Round 2: As most of the tasks, are designed in a general manner to be applicable to all use cases, we perform one more round of evaluation of the created models with only one PET example describing a claim examination[9]. The process comprises 6 tasks, 1 decision point, and involves 2 participants (see Fig. 8). This time, the redesign tasks are defined as follows:

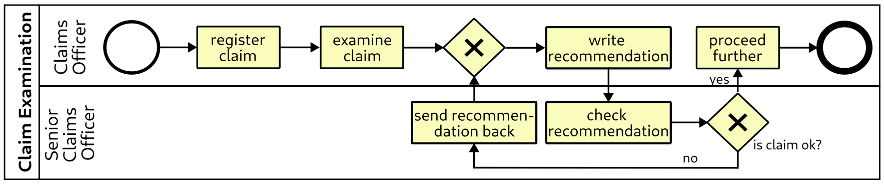


Fig. 8. Claim Handling Process

[9] Process description: https://github.com/com-pot-93/campc/blob/main/pet_examples/process_descriptions/3_3.txt

(a) add one task "login to the system" after the second task;
(b) delete the first task in the model;
(c) add an alternative branch with the task “claim is examined by senior officer” for the second task;
(d) make examination claim tasks parallel;

Table 2. Prompt Selection Assessment: Round 2 (1 Model)

	Prompt B		Prompt C		Prompt D		Prompt E	
	syntactic	semantic	syntactic	semantic	syntactic	semantic	syntactic	semantic
(a)	1	0	1	0	1	0	1	0
(b)	1	1	1	0	1	1	1	1
(c)	1	0	1	0	1	1	1	1
(d)	1	1	1	1	1	1	1	1
sum	4	2	4	1	4	3	4	3

Prompt C shows the lowest performance. Also prompt B generates a limited number of semantically correct models (see Tab. 2). Using prompt B, it is also impossible to achieve the correct syntactic structure of the model, as soon as redesign tasks attempt to integrate elements that were not previously introduced in the model, as it has no access to detailed rules for the desired output format of the model. Both prompt D and E yield same results, even though prompt E requires the largest number of input tokens for to perform the task.

The comparable performance of prompts D and B (only 1 semantically correct model less) and prompts E and C (2 semantically correct models less), indicate that by employing a process model rather than a textual process description as input can result in better outcomes. The ability of prompt B and C to generate a rather high amount of syntactically correct models without detailed description of the desired output format suggests a creditable self-learning capability of the LLM, enabling redesign of the model solely based on the structure of the input model itself.

The quality of the refined model does not only depend on supplementary information but also on the provided redesign task. Based on two rounds of iterations, it can be seen that it is better to refer to the tasks by their names rather than by their sequence numbers to achieve better results. Also, when adding parallel or alternative branches, it is important to not only define where the branch starts, but also to explicitly mention where the branch should end. Overall, we consider Prompt D as most suitable for the user study.

Limitations: The prompt selection evaluation faces several threats to validity that could affect the generalizability, and fairness of the results. Primarily, the range of BPMN constructs investigated in this work is limited and does not include elements, such as pools, lanes, and specialized gateways. Secondly, the (Round 1) evaluation is conducted on a small dataset of only seven process models, which have a rather simplistic nature. This limitation restricts the ability to generalize findings to more complex and diverse processes. In the (Round 2) evaluation, only one model is used, making it unclear whether performance differences are caused due to the use of activity labels individually for this particular process model or other contextual factors related to the LLM' and prompt's design. Additionally, the prompts themselves may introduce ambiguity due to used formulations. Potential biases in the design of synthetic tasks, combined with a subjective evaluation of correctness, further threaten the validity and overall robustness of the findings.

4 User Study

To reduce the influence of biases and limitations when evaluating the quality of models created and redesigned by LLMs, a survey involving domain experts from the manufacturing sector with extensive knowledge of the selected use case and process modelers, i.e., students and professionals with different modeling backgrounds is conducted. This direct interaction with participants helps to assess the real usability and effectiveness of the chatbot beyond quantitative assessments. Insights into how users perceive and interact with the LLMs to generate content can provide valuable information for further improvements. Participants are asked to create a process model for a use case from manufacturing, utilizing natural language via a conversational user interface. No instructions, tutorials, or additional supplementary materials are given to the participants to avoid influencing their behavior during interaction with the chatbot.

Participant background: The survey includes 10 participants categorized into domain experts and process modelers. All respondents are familiar with graphical modeling languages such as UML, ER, or BPMN. Process modelers (5 out of 10 respondents) are considered proficient, having applied modeling languages in multiple industry projects, while domain experts are only slightly familiar with modeling languages through books or individual projects.

Use case: The use case represents a genuine manufacturing process ensuring the automated production and inspection of GV12 valve-lifters to maintain quality standards, with a focus on detecting chip formation on the workpiece surface. A batch of workpieces is automatically produced and inspected to ensure the quality of each produced piece. To facilitate efficient monitoring and decision-making throughout the production cycle the process includes data collection, compression, and analysis.

Conversational interface: In order to generate the model, we employ the prompt along with one of the textual representations from [23]. To update the model, we utilize the prompt designed and evaluated in Sect. 3. As background LLM, we select GPT-4 due to its superior performance compared to other LLMs. A user interacts with the LLM using natural language, and the LLM returns a model created in the selected representation. It is important to mention that in an integrated prototype during graphical model generation using LLMs out of text, the textual representation of the model is not visible to a user. In the experiment, only flow objects as start and end events, tasks, exclusive and parallel gateways and sequence flows are considered. All respondents create an initial model and subsequently have the opportunity to perform changes in up to 3 iterations to the model. To prevent any influence on the decision making process and behavior of the participants, they are unaware of the limitation of 3 attempts.

User study results[10]**:** To assess the process models created and redesigned by the LLM in interaction between user and chatbot, the participants are presented questions about their satisfaction with the chatbot, the correctness and completeness of the models, labeling and layouting, and visual representation of the model [11]. Since only half of the participants are experienced process modelers, we first aim to investigate whether an association between modeling experience and the level of users' satisfaction exists (see Fig. 9 (a)). Both variables are categorical, each comprising five levels (see Fig. 9 (c)).

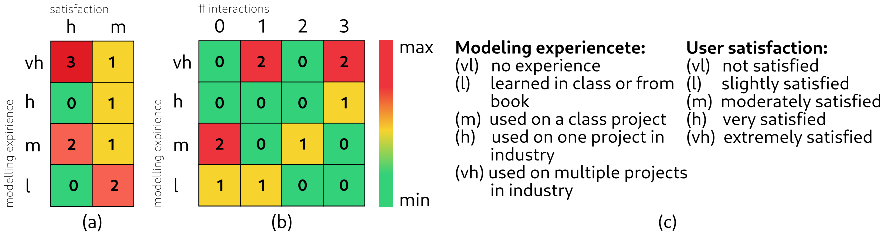


Fig. 9. User Study Results: Modeling Experience, User Satisfaction and # Interactions

It is expected that individuals with less modeling experience are more likely to be satisfied with the final model, while those with more experience are more likely to express dissatisfaction. However, based on the distribution of satisfaction levels among participants according to their experience we can observe that participants with both low (l) and high (h) levels of modeling experience achieve a moderate level of satisfaction (m) during chatbot interaction. Meanwhile, individuals with moderate (m) and very high (vh) modeling experience reach both moderate (m) and high (h) levels of satisfaction. Given the limited sample size, no clear relationship between these three variables can be seen. At the same time, there is not enough evidence to definitively say that there is no association between them. Given that 30% of all participants expressed satisfaction with the initial model, another 30% were satisfied with the model obtained after the first redesign, and only 40% engaged in 2 or 3 iterations furthermore, we move forward to investigate the relationship between the level of modeling experience and the number of iterations performed by participants to refine the model (see Fig. 9 (b)). We assume that participants with greater modeling experience tend to redesign the model more frequently than those with lower levels of experience.

Generally, we can say that participants with very high (vh) and high (h) experience are more likely to perform more interactions with the chatbot to refine the originally generated model (1 and 3 interactions), while participants

[10] The detailed study results: https://github.com/com-pot-93/campc/tree/main/user_study/study

[11] https://github.com/com-pot-93/campc/tree/main/user_study

with low (l) experience tend to have fewer interactions (0 and 1 interactions). There seems to be a relationship between experience level and the number of interactions, where participants with higher experience tend to communicate with the chatbot more intensively. However, due to the small sample size, these observations should be interpreted cautiously.

General observations: In general, among the 10 respondents, the lowest level of satisfaction with the final model was *moderate* (m), while the highest level was *very satisfied* (h). 90% of all participants assert that they have obtained a correct model (i.e., consistent with the provided process description), with 70% also claiming that their models are complete (i.e., all the tasks from the provided process description are included in the model). Also, 7 out of 10 respondents confirm that generated models are structurally correct (i.e., consistent with the BPMN 2.0 standard). Only 1 out of 10 respondents mentions to have observed hallucinations, i.e., extra modeling elements provided by LLM that are not included in the process description.

9 out of 10 participants consider the labeling of the final model as appropriate, indicating that task, gateway, and event labels are easily comprehensible. Comparable to model satisfaction, the lowest level of satisfaction with the visual representation of the final model was *moderate* (m), while the highest level was *very satisfied* (h). However, only 60% of all respondents expressed satisfaction with the auto-layouting and with the set of pre-selected BPMN elements (start and end events, tasks, exclusive and parallel gateways, and sequence flow).

Prompting style: When creating the initial models, 8 out of 10 users provided regular "story-like" text descriptions. Only 2 users attempted to operate with tasks and keywords. Initially proposed redesign tasks referred to the insertion and deletion of tasks, as well as parallel and conditional embedding (i.e., AP1, AP2, AP9, and AP10 change patterns [39]). However, none of the participants involved in model redesign utilized insertion, deletion of tasks, and conditional embedding. They primarily referred to parallelization (A9). In addition to AP9, participants also mentioned change patterns such as the replacement of elements, loop embedding, and changing conditions (AP4, AP8, and AP13 [39]).

The fact that the participants did not utilize simple change patterns may indicate that the LLM can successfully detect atomic modeling elements (tasks) and their sequence in the process. However, it struggles to identify more complex constructs related to the relationships between these elements. The achievement of a relatively high level of satisfaction during model creation, despite the fact that the utilized redesign tasks differ from those defined by participants, indicates that the prompt engineering was successful.

Our synthetic redesign statements were designed in a general manner, referring to the tasks by their sequence numbers. However, all participants found it easier to mention the task labels. This is possibly because the uniqueness of the task labels led to less misunderstanding for humans. During prompt selection, we add only one redesign task per call. It turns out that all participants also use this strategy, focusing on one change at a time. Adding only one redesign task per call seems to help maintain clarity, reduce errors, and simplify debugging.

Limitations: Given the relatively small sample size, the study results should be interpreted with caution. With half of the respondents identified as process modelers and the other half as domain experts, there is a potential impact of response biases or misinterpretation of questions. Moreover, factors like the length and complexity of the survey could impact the involvement and the responses accuracy of respondents. Also, prior user experiences and expectations can influence not only their interaction with the LLM but also their perception of the generated results. Furthermore, the variability caused by the probabilistic nature of the LLM can lead to some issues related to reproducibility (i.e., the results of the user study referring to the quality of the generated models cannot be consistently replicated).

5 Related Work

Several studies address the communication gap between domain experts and process modelers, e.g., [30,31,11,28,32]. They can be distinguished into the following two strategies, i.e., a) developing specific guidelines and recommendations and b) designing specific systems, tools, and notations for the modeling process by b1) requiring the user to adapt to predefined input formats and system rules or b2) enabling users to interact with the system using a familiar way of communication (i.e., natural language). Several works propose (a) recommendations and guidelines for different labeling styles and their use in process modeling practice [30], for changing a process model to a behavior-equivalent and more understandable model [31], and for having more efficient and effective interactions during model development [11]. Examples of approach (b.1) include a computer-based questioning system called “Process Interviewer” [28], BPMN-SBVR business vocabularies and rules converters [32], an interactive tabletop interface with tangible building blocks [22], and the design of simplified BPMN to reduce the difficulty domain experts face in learning and understanding other notations [36]. Approach b.2 can benefit from the shift in Business Process Management caused by the advancements in NLP and GenAI. This shift focuses on intelligent decision-making, NLP, and increased human-computer interaction, transforming classical BPM systems into AI-augmented Business Process Management systems [17]. These systems become conversationally actionable, meaning they can proactively communicate with human agents about process-related actions, goals, and intentions using natural language [12]. This interaction can be enhanced via the integration of intelligent chatbot functions for improved communication within the BPM framework, promoting collaboration [17]. The systems can lead conversations in a multi-turn nature, considering context and incorporating utterances from previous turns to achieve a higher degree of user engagement [9]. Currently, as mentioned in [21,8,4,37,18], there is an increasing interest in the potential benefits for the entire BPM domain arising from employing LLMs, particularly in process model generation. For instance, [6] proposes extracting process elements and relations using prompts with varying levels of pre-knowledge. In [15], the generation of an entire model with a specific level of abstraction is presented. Additionally, [25] generates complete BPMN models using LLM and POWL (Partially Ordered Workflow Language). [14] utilizes the JSON format to enhance LLMs' ability to generate not only BPMN models, but also Entity-Relationship (ER) and UML class diagrams. However, most existing approaches focus solely on single-time interactions, where the user is able to receive a final artifact from the system, but is not able to adjust it. So far, the multi-turn conversational capabilities of LLMs for process model generation have received little attention and have not yet been thoroughly explored in the Business Process Management domain.

6 Conclusions

In this work, we explore whether LLM-based chatbots can effectively support domain experts during the redesign of process models in continuous interaction via a conversational user interface to overcome the communication gap between domain experts and process modelers. The continuous interaction is based on redesign tasks of the models. To this end, we conducted a prompt design experiment for process model redesign tasks. The selected prompt was then applied in a user study with domain experts and process modelers on a manufacturing process model. It can be seen that the quality of a model redesign is highly dependent not only on a prompt design but also on how the redesign task is described and the complexity of the task itself. 90% of all participants assert that they have produced a correct model, meaning it is consistent with the provided process description. Additionally, 70% of participants claim that their models are complete, including all expected tasks from the process description, and structurally correct, adhering to the BPMN 2.0 standard. Future research will focus on two primary directions. The first will explore multiple change patterns and more complex datasets to address the increasing complexity of real-world scenarios. The second direction will emphasize evaluating and integrating knowledge about user behavior to improve the quality of human-chatbot communication, better meeting the needs of domain experts. Additionally, observing this communication as a learning process for domain experts may help develop their modeling skills and foster *process thinking* through active engagement in process model creation.

References

1. Ali, M., Fromm, M., Thellmann, K., Rutmann, R., Lübbering, M., Leveling, J., Klug, K., Ebert, J., Doll, N., Buschhoff, J.S., et al.: Tokenizer choice for llm training: Negligible or crucial? arXiv preprint arXiv:2310.08754 (2023)
2. Azevedo, L.G., Rodrigues, R.D.A., Revoredo, K.: BPMN model and text instructions automatic synchronization. In: Enterprise Inf. Syst. pp. 484–491 (2018)
3. Bakker, M., Chadwick, M., Sheahan, H., Tessler, M., Campbell-Gillingham, L., Balaguer, J., McAleese, N., Glaese, A., Aslanides, J., Botvinick, M., et al.: Finetuning language models to find agreement among humans with diverse preferences. Advances in Neural Information Processing Systems **35**, 38176–38189 (2022)

4. Beheshti, A., Yang, J., Sheng, Q.Z., Benatallah, B., Casati, F., Dustdar, S., Motahari-Nezhad, H.R., Zhang, X., Xue, S.: Processgpt: Transforming business process management with GenAI. In: Web Services. pp. 731–739 (2023)
5. Bellan, P., van der Aa, H., Dragoni, M., Ghidini, C., Ponzetto, S.P.: PET: an annotated dataset for process extraction from natural language text tasks. In: Business Process Management Workshops. pp. 315–321 (2022)
6. Bellan, P., Dragoni, M., Ghidini, C.: Extracting business process entities and relations from text using pre-trained language models and in-context learning. In: Enterprise Design, Operations, and Computing. pp. 182–199 (2022)
7. Beverungen, D.: Exploring the interplay of the design and emergence of business processes as organizational routines. Bus. Inf. Syst. Eng. **6**(4), 191–202 (2014)
8. Busch, K., Rochlitzer, A., Sola, D., Leopold, H.: Just tell me: Prompt engineering in business process management. In: Enterprise, Business-Process and Information Syst. Modeling. pp. 3–11 (2023)
9. Casciani, A., Bernardi, M.L., Cimitile, M., Marrella, A.: Conversational systems for ai-augmented business process management. In: Research Challenges in Information Science. pp. 183–200 (2024)
10. DaSilva, C.M., Trkman, P.: Business model: What it is and what it is not. Long Range Planning **47**(6), 379–389 (2014)
11. Doren, A., Markina-Khusid, A., Cotter, M., Dominguez, C.: A practitioner's guide to optimizing the interactions between modelers and domain experts. In: Systems. pp. 1–8 (2019)
12. Dumas, M., et al.: AI-augmented business process management systems: A research manifesto. ACM Transactions on Management Inf. Syst. **14**, 1 – 19 (2022)
13. Dumas, M., Rosa, M.L., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management. Springer (2013)
14. Fill, H., Fettke, P., Köpke, J.: Conceptual modeling and large language models: Impressions from first experiments with chatgpt. Enterp. Model. Inf. Syst. Archit. Int. J. Concept. Model. **18**, 3 (2023)
15. Grohs, M., Abb, L., Elsayed, N., Rehse, J.: Large language models can accomplish business process management tasks. CoRR **abs/2307.09923** (2023)
16. Gutschmidt, A., Lantow, B., Hellmanzik, B., Ramforth, B., Wiese, M., Martins, E.: Participatory modeling from a stakeholder perspective: On the influence of collaboration and revisions on psychological ownership and perceived model quality. Software and Systems Modeling **22**, 1–17 (08 2022)
17. Hildebrand, D., Rösl, S., Auer, T., Schieder, C.: Next-generation business process management (bpm): A systematic literature review of cognitive computing and improvements in bpm (05 2024)
18. Jessen, U., Sroka, M., Fahland, D.: Chit-chat or deep talk: Prompt engineering for process mining. CoRR **abs/2307.09909** (2023)
19. Jieon Lee, D.L., gil Lee, J.: Influence of rapport and social presence with an ai psychotherapy chatbot on users' self-disclosure. International Journal of Human–Computer Interaction **40**(7), 1620–1631 (2024)
20. Jin, H., Han, X., Yang, J., Jiang, Z., Liu, Z., Chang, C., Chen, H., Hu, X.: LLM maybe longlm: Self-extend LLM context window without tuning. CoRR **abs/2401.01325** (2024)
21. Kampik, T., Warmuth, C., Rebmann, A., Agam, R., Egger, L.N.P., Gerber, A., Hoffart, J., Kolk, J., Herzig, P., Decker, G., van der Aa, H., Polyvyanyy, A., Rinderle-Ma, S., Weber, I., Weidlich, M.: Large process models: Business process management in the age of generative AI. CoRR (2023)

22. Kannengiesser, U., Oppl, S.: Business processes to touch: Engaging domain experts in process modelling. vol. 1418 (09 2015)
23. Klievtsova, N., Benzin, J., Kampik, T., Mangler, J., Rinderle-Ma, S.: Conversational process modelling: State of the art, applications, and implications in practice. In: Business Process Management Forum. pp. 319–336 (2023)
24. Klievtsova, N., Mangler, J., iik, T., Benzin, J.V., Rinderle-Ma, S.: How can generative ai empower domain experts in creating process models? In: Wirtschaftsinformatik (2024), (accepted)
25. Kourani, H., Berti, A., Schuster, D., van der Aalst, W.M.P.: Promoai: Process modeling with generative ai. ArXiv **abs/2403.04327** (2024)
26. Leopold, H., van der Aa, H., Pittke, F., Raffel, M., Mendling, J., Reijers, H.: Searching textual and model-based process descriptions based on a unified data format. Software and Systems Modeling **18** (04 2019)
27. Leopold, H., Mendling, J., Polyvyanyy, A.: Supporting process model validation through natural language generation. IEEE Trans. Software Eng. **40**(8), 818–840 (2014)
28. Ley, D.: Approximating process knowledge and process thinking: Acquiring workflow data by domain experts. 2011 IEEE International Conference on Systems, Man, and Cybernetics pp. 3274–3279 (2011)
29. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. ACM CSUR **55**(9) (2023)
30. Mendling, J., Reijers, H., Recker, J.: Activity labeling in process modeling: Empirical insights and recommendations. Information Systems **35**(4), 467–482 (2010)
31. Mendling, J., Reijers, H., Aalst, W.: Seven process modeling guidelines (7pmg). Information and Software Technology **52**, 127–136 (02 2010)
32. Mickeviciute, E., Butleris, R., Gudas, S., Karciauskas, E.: Transforming bpmn 2.0 business process model into sbvr business vocabulary and rules. Inf. Technol. Control. **46**, 360–371 (2017)
33. Mursyada, A.: The role of business process modeling notation in process improvement: A critical review. Advanced Qualitative Research (2024)
34. Odeh, Y.: Bpmn in engineering software requirements: An introductory brief guide. In: International Conference on Information Management and Engineering (2017)
35. Rinderle-Ma, S., Reichert, M., Weber, B.: On the formal semantics of change patterns in process-aware information systems. In: ER. pp. 279–293 (2008)
36. Solís-Martínez, J., Espada, J.P., Pelayo G-Bustelo, B.C., Lovelle, J.M.C.: Bpmn musim: Approach to improve the domain expert's efficiency in business processes modeling for the generation of specific software applications. Expert Systems with Applications **41**(4, Part 2), 1864–1874 (2014)
37. Vidgof, M., Bachhofner, S., Mendling, J.: LLMs for business process management: Opportunities and challenges. In: BPM Forum. pp. 107–123 (2023)
38. Wang, B., Wang, C., Liang, P., Li, B., Zeng, C.: How llms aid in uml modeling: An exploratory study with novice analysts. ArXiv **abs/2404.17739** (2024)
39. Weber, B., Reichert, M., Rinderle-Ma, S.: Change patterns and change support features - enhancing flexibility in process-aware information systems. Data Knowl. Eng. **66**(3), 438–466 (2008)
40. Weber, B., Zeitelhofer, S., Pinggera, J., Torres, V., Reichert, M.: How advanced change patterns impact the process of process modeling. CoRR **abs/1511.04060** (2015)