How Are LLMs Used for Conceptual Modeling? An Exploratory Study on Interaction Behavior and User Perception

Syed Juned Ali[1], Iris Reinhartz-Berger[2], and Dominik Bork[1]

[1] TU Wien, Business Informatics Group, Vienna, Austria
{syed.juned.ali, dominik.bork}@tuwien.ac.at
[2] University of Haifa, Haifa, Israel iris@is.haifa.ac.il

Abstract. Large Language Models (LLMs) have opened new opportunities in modeling in general, and conceptual modeling in particular. With their advanced reasoning capabilities, accessible through natural language interfaces, LLMs enable humans to deepen their understanding of different application domains and enhance their modeling skills. However, the open-ended nature of these interfaces results in diverse interaction behaviors, which may also affect the perceived usefulness of LLM-assisted conceptual modeling. Existing works focus on various quality metrics of LLM outcomes, yet limited attention is given to *how users interact with LLMs* for such modeling tasks. To address this gap, we present the design and findings of an empirical study conducted with information systems students. After labeling the interactions according to their intentions (e.g., Create Model, Discuss, or Present), and representing them as an event log, we applied process mining techniques to discover process models. These models vividly capture the interaction behaviors and reveal recurrent patterns. We explored the differences in interacting with two LLMs (GPT 4.0 and Code Llama) for two modeling tasks (use case and domain modeling) across three application domains. Additionally, we analyzed user perceptions regarding the usefulness and ease of use of LLM-assisted conceptual modeling.

Keywords: Large Language Model · Domain Modeling · UML · Process Mining.

1 Introduction

In recent years, the advent of Large Language Models (LLMs) has revolutionized the landscape of natural language processing and machine learning. Trained on vast amounts of text data, LLMs exhibit remarkable proficiency in understanding and generating human-like text. They have proven to be effective tools across various tasks [29], including translation [27,18], text summarization [28,24], sentiment analysis [21,2,9] and recommender systems [19,15]. Particularly noteworthy is their significant impact on software engineering, where they excel in tasks such as program synthesis from natural language specifications, code completion, debugging, and documentation generation [11,16].

The exploration of LLMs' potential to enhance modeling tasks has only recently gained attention, as evidenced by emerging research, e.g., [6,12,5,7,3,17,8]. Most of these works focus on the accuracy, utility and other quality metrics of the LLM-produced models. However, to the best of our knowledge, no study has investigated how modelers interact and perceive their interactions with LLMs for conceptual modeling tasks. These interactions can be influenced by factors such as the specific LLM used, the application domain, and the nature of the modeling task itself. Thus, our objective is to discover the process of interacting with LLMs and identify recurrent patterns that can inform 'best practices' for LLM-assisted conceptual modeling.

In this paper, we report on the design and results of an empirical study with 76 undergraduate information systems students. The students submitted the course assignments in 39 groups (of one or two students). The assignment used for the study required use case modeling with UML use case diagrams and domain modeling with UML class diagrams. The groups were first expected to interact with an LLM and then, if needed, to manually improve their models. Groups were randomly assigned to interact with either GPT 4.0 or Code Llama 34B Instruct in one of three application domains. To facilitate data collection without disclosing the LLM's identity, we developed a dedicated web application capable of logging all interactions between the groups and the LLMs. Finally, the students' perceptions were collected via a standard usefulness and ease of use questionnaire, in which the students were also requested to list the main positive and negative aspects they experienced.

The rest of the paper is structured as follows. Sections 2 and 3 elaborate on the experimental design and analysis procedure, respectively. Sections 4 and 5 present and discuss the results. Finally, Section 6 reviews related work, and Section 7 concludes and refers to future research directions.

2 Experimental Design

2.1 Goals, Research Questions, and Independent Variables

The goal of this study is to analyze modeler interactions with LLMs, for the purpose of identifying the underlying intentions and patterns, as well as exploring user perceptions on interacting with LLMs for conceptual modeling tasks. To this end, we identified the following two main research questions:

[RQ1] How do users interact with LLMs for conceptual modeling tasks?
[RQ2] How do users perceive the usefulness and ease of use of those interactions?

Considering that responses to these questions can be influenced by various factors, we identify three independent variables that potentially impact the modeling process and experience. These variables are:

- **LLM** –We investigate GPT-4 and Code-Llama 34B Instruct;
- **Application Domain** – We investigate NPA (Nature & Park Authority) for nature & archaeological sites management, R4A (Rating for All) for viewership data analytics, and PTr (Perfect Trip) for tourism management;

- **Task** – We investigate UCD (use case modeling with Use Case Diagrams) and CD (domain modeling with Class Diagrams).

Accordingly, we divided the first research question into the following sub-questions:

[RQ1.1] What are the recurrent interaction behaviors (i.e., intentions and patterns)?
[RQ1.2] To what extent do the recurrent interaction behaviors vary across different LLMs?
[RQ1.3] To what extent do the recurrent interaction behaviors vary across different application domains?
[RQ1.4] To what extent do the recurrent interaction behaviors vary across different modeling tasks?

We further refined the second research question into the following sub-questions[1]:

[RQ2.1] What are the overall perceived usefulness and ease of use users experience when utilizing LLMs for conceptual modeling?
[RQ2.2] To what extent do the perceived usefulness and ease of use vary across different LLMs?
[RQ2.3] To what extent do the perceived usefulness and ease of use vary across different application domains?

2.2 Settings and Objects

The rationale behind selecting GPT-4 and Code-Llama (referred to as Llama henceforth) lies in their distinct capabilities and performance metrics. GPT-4, a large multi-modal, general-purpose model capable of processing both image and text inputs, has demonstrated remarkable performance across various human-designed benchmarks [23]. Studies, such as [1], have highlighted GPT-4's superior performance, often outscoring a significant majority of human test takers. Llama, on the other hand, encompasses a range of LLMs of varying sizes developed by Meta AI [2], specifically designed for coding tasks. Llama stands out for its state-of-the-art performance among open models, robust infilling capabilities, support for extensive input contexts, and the ability to execute programming instructions without the need for prior training [14]. Notably, its fine-tuned 70 billion parameters variant has shown exceptional performance in coding tasks, outperforming GPT-4 in benchmarks such as HumanEval [23] [3].

Although GPT-4 features a larger context window of 32,000 tokens[4] compared to Llama's 16,000 tokens, allowing it to retain more information across interactions, , our decision to include both models in the study is driven by their

[1] We have no dedicated sub-question regarding differences related to the modeling tasks, as the participants could work on them interwinedly.

[2] `https://www.meta.ai/`

[3] `https://llama.meta.com/code-llama/`

[4] 1 token ≈ 0.75 words.

Table 1: Descriptive statistics of application domains

Artifact	Element	NPA	R4A	PTr	Avg	Std
UCD	Use Case	11	14	12	12.33	1.25
	Actor	6	6	6	6.00	0.00
	Association	9	12	12	11.00	1.41
	Dependency	3	4	5	4.00	0.82
CD	Class	14	13	12	13.00	0.82
	Enumeration	2	2	1	1.67	0.47
	Attribute	36	26	30	30.67	4.11
	Operation	2	2	2	2.00	0.00
	Generalization	2	2	2	2.00	0.00
	Association	8	7	8	7.67	0.47
	Association Class	2	2	3	2.33	0.47
Description	Length in GPT-4 tokens	846	856	855	852.33	4.49
	Length in Llama tokens	943	945	943	943.66	0.94

respective strengths — GPT-4's versatility in general-purpose tasks and Llama's specialization in coding-related tasks.

We further selected three application domains that are likely familiar to humans and LLMs: nature & archaeological sites management (NPA), viewership data analytics (R4A), and tourism management (PTr). Despite their differences, we took care to ensure that their expected models were of comparable size and complexity, as depicted by the descriptive statistics in Table 1. The entire experimental material is provided in our online supplementary material[5].

2.3 Modeling Tasks

The participants were asked to perform two modeling tasks using UML notation: use case modeling employing UCD and domain modeling utilizing CD. This choice is supported by their widespread adoption as well as their representation of distinct yet complementary aspects. The participants were instructed to engage with the LLM until they were satisfied with the results or opted to skip further refinement. Subsequently, they had the opportunity to enhance the models within a modeling tool. This process resulted in two distinct outcomes for each task: a *'DRAFT'* model, solely derived from interactions with the LLM, and a *'FINAL'* model, refined through additional manual adjustments in a modeling tool. Grading primarily focused on the 'FINAL' outcomes (85%). To incentivize engagement, interactions with LLMs accounted for 15% of the grade. Notably, the 'DRAFT' models did not factor into the grading.

[5] Online supplementary material: `https://zenodo.org/records/13513891`

2.4 Instrumentation and Data Collection

In order to collect the participants' interactions without revealing the LLM they are using, we designed a web application built using Streamlit [6], which is an open-source Python library for creating web applications. Our application permits up to 100 prompts per user (who can include one or two participants, see Section 2.5 for more details). This limit aimed to encourage participants to generate meaningful prompts for their tasks.

The interface design is similar to those of existing chatbots, allowing users to input their prompts and receive modeling artifacts in response. A screenshot of the interface is included in the experimental material[5]. The application also presents users with the number of prompts they have used and the number of remaining prompts. Additionally, it enables the research team to download an *interaction log* with the following fields: User ID, Input (the user prompt), Response (the modeling artifacts), and the Prompt Number (within user ID).

2.5 Participants and Experimental Design

The experiment took place in the academic year of 2024 in a second-year undergraduate course on 'Design and Implementation of Information Systems'. The course focused on object-oriented modeling with UML. The students were enrolled in a three-year BSc program in Information Systems. They were already knowledgeable in programming in general and in object-oriented programming with Java in particular. The tasks were part of the course assignments and were mostly submitted in groups of two students, with only two exceptions where students submitted individually. Table 2 summarizes the experiment design and the division of the 39 groups along the LLMs and the application domains. Each group was assigned to a single application domain and a single LLM, and had to perform both modeling tasks.

The experiment comprised three stages: (*i*) a 30-minute tutorial providing a brief introduction to LLMs, prompt engineering, and a demonstration of the application the students were required to use for their assignment; (*ii*) a two-week window during which the students were asked to complete and submit the artifacts of the two modeling tasks; and (*iii*) a questionnaire (see Section 4.2 for the questions) aimed at assessing each student's perceived usefulness and ease of use when interacting with the LLM for both modeling tasks.

Table 2: Number of groups per experimental category

Domain ↓ LLM →	**Llama**	**GPT-4**	**Total**
NPA	8	7	15
R4A	7	6	13
PTr	5	6	11
Total	**20**	**19**	**39**

[6] `https://docs.streamlit.io/`

Table 3: Intentions used for labeling interaction prompts

Intention	Description
Create Model	Relates to the creation of a (potentially partial) model. This prompt does not refer to previous responses.
Update Model	Relates to the update of a (potentially partial) model. This prompt refers to previous responses, e.g., for correcting or clarifying previous outcomes.
Create List	Relates to the creation of a list of modeling elements, such as classes, use cases and associations. This prompt does not refer to previous responses.
Update List	Relates to the update of a list of modeling elements. This prompt refers to previous responses.
Present	Relates to the presentation of the response in a given format (e.g., XMI).
Explain	Asks the LLM to explain certain parts of previous responses. The purpose of Explain prompts is to understand and not update an outcome.
Discuss	Asks the LLM to discuss possible solutions (presented either explicitly or implicitly), appearing in previous prompts or responses. Differently from Explain, Discuss has an implicit intention for updating an outcome.

3 Analysis Procedure

Interaction Labeling. After downloading the log with 541 interactions in total, we explored the user prompts and identified seven intention types. Table 3 describes and explains the identified intentions in detail. They refer to model development operations (*'Create Model'*, *'Update Model'*, *'Create List'*, *'Update List'*), model presentation operations (*'Present'*), and explanatory operations (*'Explain'*, *'Discuss'*). Each interaction underwent manual labeling by two of the three conducting researchers, with each researcher responsible for labeling two-thirds of the interactions. While doing so, we also identified the task type (UCD or CD) of each interaction. Initially, our agreement level for intention labeling stood at 64.3%. To ensure consistency and accuracy, we engaged in multiple discussion sessions to refine the definitions of the various labels, eventually achieving full agreement on the labeling outcome.

Following the labeling phase, we encountered prompts that combined multiple intentions, requiring the splitting of rows in the log file. Additionally, we identified eight prompts lacking clear intentions, comprising only descriptions without actionable requests, which we consequently omitted. After these preprocessing steps, we were left with 566 interactions for analysis.

Behavior Extraction. We treated the interactions log as an event log and employed the process mining tool Disco[7] to: (*i*) discover the interaction process; and (*ii*) extract recurrent interaction patterns. We considered the combination of the user ID (corresponding to groups 1 to 39) and the task type (UCD or CD) as the case ID. We further added information on the LLM (GPT-4, Llama) and the application domains (NPA, R4A, PTr) of each group to check differences in the interaction behavior based on the independent variables.

[7] https://fluxicon.com/disco/

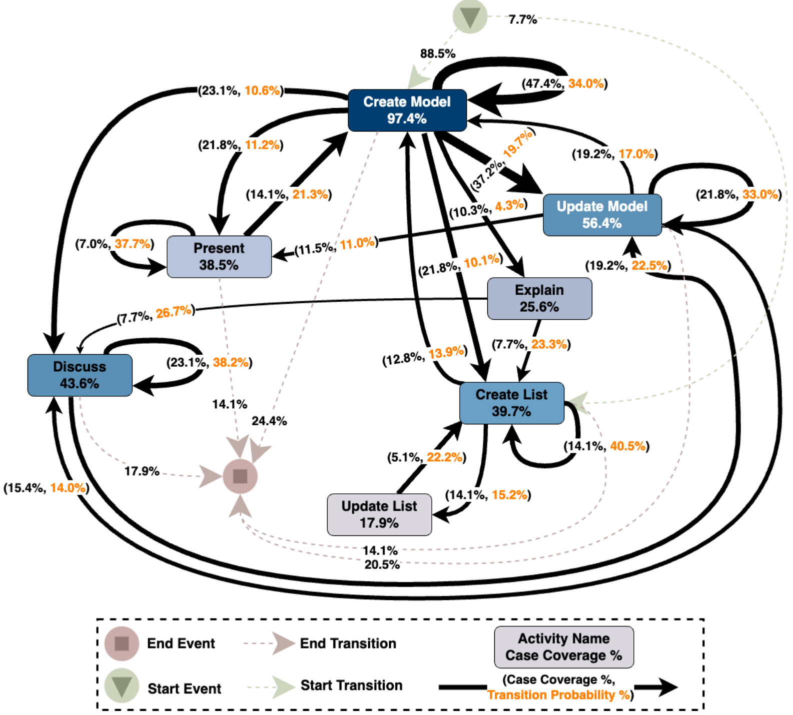

Fig. 1: Discovered interaction process model

4 Results

Next, we elaborate on the results of our study and answer the research questions.

4.1 Results for RQ1: Interaction Behavior

Overall Behavior Results (RQ1.1). In the discovered interaction process model (see Fig. 1), nodes depict intentions while edges depict transitions between intentions, i.e., appearance in sequential prompts. The percentages displayed on the nodes indicate case coverage, i.e., the proportion of cases employing an intention. On the edges, the black value on the left indicates the case coverage of transitions, while the orange value on the right refers to individual intention instances (rather than grouped by cases), showing the probability of proceeding with that transition from the source intention. Less traversed paths with a case coverage of ≤ 10% and a probability of ≤ 20% are omitted for the sake of comprehensibility. The interaction log is provided with the experimental material[5].

Model Development Operations. In nearly all cases (97.4%), 'Create Model' was employed, and in 88.5% of cases, it initiated the interaction with the LLM. These statistics align with expectations given the nature of tasks requiring model creation. Notably, the most frequent transition from 'Create Model', observed in 47.4% of cases, is to another 'Create Model.' Moreover, 'Create Model' has a probability of 34.0% to follow 'Create Model', compared to

'Update Model' which has a lower probability of 19.7% and covers fewer cases (37.2%). This preference for initiating new models over updating existing ones in specific scenarios can be attributed to participants' desire for a fresh start and the potential for clearer conceptualization. This inclination contrasts with the operation of modifying existing models, which may involve navigating and adapting previous decisions or structures. Another explanation could be the limited capacity of the LLMs to retain information over interactions depending on their maximum context window, leading to *forgetting* of information by the LLM and consequently generating unsatisfactory models that are not worth updating.

The results also reveal that 'Update Model' is a common intention, appearing in 56.4% of cases, with subsequent updates occurring in 21.8% of cases (vs. 19.2% cases that start over with 'Create Model'). This observed behavior implies an iterative process of model refinement, potentially influenced by feedback from the LLM or as additional information becomes available.

We further observed interaction paths in which lists are used, before or after using models. These paths were less frequent, accounting for 39.7% case coverage for 'Create List' and 17.9% for 'Update List.' These intentions primarily revolved around listing model elements, such as classes, associations, use cases, and actors. In some cases (7.7%), 'Create List' was the initial intention. However, it typically followed 'Create Model' (in 21.8% of the cases), potentially to assess the suggested model by listing certain modeling elements. Once lists were established, they were re-created, modified, or utilized to (re)create a model. List re-creating and modification had the same case coverage of 14.1%, yet the first exhibited a higher probability of 40.5% compared to list modification with a probability of 15.2%. In 12.8% of the cases, list creation was followed by a subsequent model creation with a transition probability of 13.9%.

Explanatory Operations. We observed that the use of explanatory operations was moderate, with 'Discuss' at case coverage of 43.6% and 'Explain' at 25.6%. Discussions embody a collaborative aspect, where modelers utilize the LLM outputs for deep exploration or decision-making between alternatives. The repeated use of 'Discuss' (with a case coverage of 23.1% and a probability of 38.2%) reflects ongoing clarification and negotiation of ideas, further indicating an active dialogue-based interaction with the LLM. A closer examination of the prompts reveals dissatisfaction with the LLM's responses in some cases. In about one-fifth of cases (19.2%), the discussion results or insights are integrated into the model through performing 'Update Model.'

'Explain' intentions aim to provide detailed insights into previous LLM responses. Through explanations, participants shed light on the reasoning behind specific outcomes, foster comprehension, and reach informed decision-making in subsequent interactions. Given that LLMs frequently offer explanations in their responses, the occurrence of 'Explain' is less frequent. Interestingly, in 23.3% of the instances, 'Explain' preceded 'Create List,' which may serve to formalize the explanation results as lists of modeling elements. In 26.7% of instances, 'Explain' is followed by 'Discuss,' potentially indicating a desire for deeper insights into the suggested options.

Table 4: Five most recurring interaction patterns

Code	**Pattern**	**Overall**	**No. of Groups (out of 39)**	**No. of Cases (out of 78)**
P_1	Model Evolution Create Model+; Update Model+	37	20	29
P_2	Partial Model Visualization Create Model+; Present+	21	14	17
P_3	Partial Model Discussion Create Model+; Discuss+	20	14	18
P_4	Discussion-based Model Updating Discuss+; Update Model+	20	12	15
P_5	Partial Model Listing Create Model+; Create List+	19	13	17

Presentation Operations. 'Present' involves visualizing or representing outcomes in importable formats. This intention was observed in 38.5% of the cases where participants sought to visualize or import the results of their modeling efforts into a modeling environment to, e.g., continue manual modeling there. The repeated sequential uses of this intention (with a probability of 37.7%) may indicate dissatisfaction with the results. Indeed, both LLMs faced difficulties in generating outcomes that could be directly imported to the modeling environment used in the course (Visual Paradigm). Additionally, in 14.1% of cases (a probability of 21.3%), 'Present' is followed by 'Create Model,' suggesting a need to (re)create a model after reviewing the previous outcomes in a specific format.

Recurrent Patterns. The process model in Fig. 1 does not cover interaction patterns, i.e., paths that go beyond immediate transitions. Hence, in Table 4, we show the top five patterns with their overall instance frequencies, group coverage, and case coverage. The '+' sign indicates potential repetitive occurrences of an intention. Note, that since it is important to relate the recurrent patterns to the quality of the created models, we also examined the top recurrent patterns of cases achieving a passing grade. Despite a slight difference in their ordering, the patterns remained the same. We leave the analysis of the impact of specific interaction behaviors on the quality of models to future research.

As expected, the most frequent pattern is *'Model Evolution'* (P_1). This pattern starts with a sequence of 'Create Model' prompts followed by a sequence of 'Update Model' prompts. This pattern captures the iterative nature of conceptual modeling where refinement and progressive enhancement are key.

The second most recurrent pattern, *'Partial Model Visualization'* (P_2), also starts with a sequence of 'Create Model' prompts, but this is followed by 'Present' prompts, which aim at visualizing the modeling outcomes in certain formats. This step precedes decision-making on subsequent actions such as updating the outcomes, completing the task, seeking for explanations, or initiating the task.

The third pattern *'Partial Model Discussion'* (P_4) indicates pauses in the model creation process to explore and discuss potential alternatives, which may be subsequently implemented in the model (e.g., using pattern P_4). The fourth pattern *'Discussion-based Model Updating'* (P_3) empowers participants to distill discussion insights into actionable items for updating the model. Finally,

Table 5: Intention distribution categorized by independent variables. Significant results are in bold with an asterisk.

	LLM (%)		**Application Domain** (%)			**Task** (%)	
Intention	**GPT-4**	**Llama**	**NPA**	**PTr**	**R4A**	**UCD**	**CD**
Create Model	28.92	35.73	37.68	33.81	28.76	32.86	33.68
Update Model	**26.47***	**12.74***	**24.63***	19.42	**10.04***	15.90	19.50
Create List	**4.41***	**19.39***	**6.28***	12.95	**21.91***	13.78	14.18
Update List	1.96	3.87	1.93	3.59	4.07	2.12	4.25
Present	7.35	12.74	**5.31***	8.63	**17.35***	12.01	9.57
Explain	7.84	3.87	2.89	7.19	6.39	4.94	5.34
Discuss	**23.03***	**11.63***	**21.25***	14.38	**11.41***	18.37	13.12
Prompts Per Case	5.36	9.05	6.90	6.31	8.4	6.90	7.23

Table 6: Ranking of top five interaction patterns across independent variables

	LLM		**Application Domain**			**Task**	
Code	**GPT-4**	**Llama**	**NPA**	**PTr**	**R4A**	**UCD**	**CD**
$\mathbf{P}_1$	Top 2	Top 1	Top 1	Top 1	Top 3	Top 1	Top 1
$\mathbf{P}_2$	Top 6	Top 3	Top 7	Top 2	Top 3	Top 4	Top 2
$\mathbf{P}_3$	Top 3	Top 8	Top 6	Top 7	Top 2	Top 4	Top 4
$\mathbf{P}_4$	Top 1	Top 28	Top 3	Top 7	Top 6	Top 7	Top 2
$\mathbf{P}_5$	Top 21	Top 2	Top 10	Top 4	Top 1	Top 2	Top 6

'Partial Model Listing' (P_5), illustrates a shift in the modeling approach where participants move from working directly with models to using lists. This is used to explore specific aspects of the model, such as structural components by listing all classes and associations. Unlike the iterative process observed in 'Model Evolution' (P_1), participants following this pattern aim to systematically break down and analyze individual elements of the model to gain a more detailed understanding and enhance their modeling outcomes.

Behavior Results by Independent Variables (RQ1.2–RQ1.4). Next, we analyzed the results according to the independent variables – LLM, domain, and task. Tables 5 and 6 present the intention distribution and the ranking of the top five patterns across the independent variables, respectively. We further conducted pairwise comparisons of intention probabilities using Z-tests to compare proportions of independent populations. For reporting the statistical significance, we applied the Bonferroni adjustment [4] to control for multiple comparisons, reducing the likelihood of false positives by providing a more rigorous criterion for significance.

Interaction Behavior with respect to LLM (RQ1.2). Overall, we noted a significantly higher average number of interactions when engaging with Llama compared to GPT-4 (9.03 vs. 5.37 per case). This observation aligns with the fact that GPT-4 currently ranks as the most proficient LLM[8]. Our findings further revealed significant differences in the use of three intentions. Specifically, 'Create List' emerged as significantly more prevalent in Llama prompts compared to those with GPT-4 (19.39% versus 4.41%, respectively). This observation can be attributed to Llama's specialization in coding tasks, which emphasizes breaking down problems into structured lists or steps, thereby encouraging users to approach tasks in a similar manner. On the other hand, 'Update Model' and 'Discuss' were significantly more common in GPT-4 interactions than in those with Llama (26.47% versus 12.74% for 'Update Model' and 23.03% versus 11.63% for 'Discuss'). This discrepancy may stem from GPT-4's broader language understanding and its tendency to facilitate dynamic exchanges and conceptual refinement, thus encouraging users to update existing models or engage in discussions more frequently.

As seen in Table 6, three of the top-five patterns appear on the top-five lists for both GPT-4 and Llama, i.e., 'Model Evolution' (P_1), 'Partial Model Discussion' (P_3), 'Discussion-based Model Updating' (P_4) for GPT-4; and 'Model Evolution' (P_1), 'Partial Model Visualization' (P_2), 'Partial Model Listing' (P_5) for Llama. In other words, GPT-4 seems to encourage more interactive and collaborative dialogues, while Llama nudges users to follow a try-and-fix approach, involving the listing of model elements and presenting partial models before continuing with model updating and refinement.

Interaction Behavior with respect to Application Domain (RQ1.3). In terms of the application domain, we conducted pairwise comparisons and discovered four significant differences, particularly between NPA and R4A: 'Create List' and 'Present' were more frequently utilized in R4A, while 'Discuss' and 'Update Model' were more common in NPA. Although the descriptive statistics in Table 1 do not offer an immediate explanation for this observation, it is possible that the participants were less familiar with the NPA domain compared to R4A, leading to more frequent discussions and model updates. In R4A tasks, the emphasis was on organizing and systematically presenting already known data.

Moreover, the top five patterns appear on the top ten lists of all three domains. This is not surprising given that the application domains are of comparable size and complexity. These findings imply that LLM-assisted modeling tools do not necessarily need to be fine-tuned for different application domains. Further research is needed to study whether and how the size and complexity of application domains affect the distribution of intentions and the most recurrent interaction patterns. This is particularly relevant for large domain descriptions exceeding the LLM's context window, as participants often used the entire textual domain description in their interactions.

Interaction Behavior with respect to Task (RQ1.4). The results further indicate that there are no statistically significant differences in the distribution of intentions or the most recurrent interaction patterns between use case modeling with UCD and domain modeling with CD. This suggests that the underlying processes and patterns of interaction remain consistent across differ-

[8] `https://www.vellum.ai/llm-leaderboard`, last accessed: 25.05.2024.

ent modeling tasks, implying that tools and techniques used for LLM-assisted modeling may be broadly applicable and versatile. However, further research is needed to validate this hypothesis and explore its implications.

Summary of RQ1 Results. From the analyses conducted, we conclude that interactions with LLMs for conceptual modeling primarily involve prompts aimed at creating and updating models, generating and modifying lists of model elements, explaining and discussing modeling alternatives or decisions, and presenting models (RQ1.1). Several recurrent patterns have emerged, supporting model evolution, partial model visualization and listing, and discussion-based model development (RQ1.1). Notably, no statistically significant differences have been observed in terms of interaction behavior across modeling tasks (RQ1.4), while a few explained differences have been noted in the frequency of certain intentions across LLMs (RQ1.2) and application domains (RQ1.3).

4.2 Response to RQ2: User Perception

Table 7 shows the perceived usefulness and ease of use of LLM-assisted modeling both overall and segmented by LLMs and application domains. Due to the iterative work on the modeling tasks, the results reflect the participants' feedback at the end of the assignment, after submitting both modeling artifacts.

Our questionnaire comprised 14 questions: six closed questions assessing *perceived usefulness*, six closed questions evaluating *perceived ease of use*, and two open-ended questions. Responses to closed questions were rated on a Likert scale from 1 (unlikely) to 7 (likely). The open-ended questions asked to list the most *negative* and *positive* aspects the participants faced while utilizing LLMs for modeling tasks. We employed GPT-4 to analyze and categorize the participants' feedback to these open questions. Furthermore, to evaluate the statistical significance of LLM and Application Domain, we performed the Kruskal–Wallis H-test [20] and show the p-values in Table 7.

Overall User Perception (RQ2.1). Overall, the results show moderate user experiences, with an average of 4.0 for perceived usefulness and 4.4 for perceived ease of use. The only statistically significant result relates to the ease of learning to operate LLMs, suggesting the potential for steep learning curves associated with LLM-assisted modeling. The borderline significant result concerning the ease of becoming skillful at using LLMs further supports this observation.

To gain a deeper understanding of user perception, we analyzed their textual feedback. The positive feedback mainly referred to: (*i*) Efficiency and speed – "it is very responsive and quick," "Fast response, response explanation;" (*ii*) Guidance and direction – "it give[s] you a way to start your solution," "Helped me build something basic to start from;" and (*iii*) Supports Creativity – "it has a unique thinking about the problem," "provides too much aspects of certain topics which help generate new ideas." The negative aspects that were widely mentioned referred to: (*i*) Lack of contextual memory and continuity – "it gives answers to the question that was just asked and does not give answers that combine all the requirements of the question," "its answers aren't based on previous

Table 7: Perceived usefulness and ease of use, overall and categorized according to LLMs and application domains. Significant results are in bold with an asterisk.

	Overall		**LLM**		**Application Domain**	
	Mean [SD]	**p-value**	**Mean [SD][1]**	**p-value**	**Mean [SD][2]**	**p-value**
			Perceived Usefulness			
Using LLMs for modeling would enable me to accomplish tasks more quickly	4.2 1.9	0.425	(4.7, 3.6) (1.9, 1.7)	**0.017***	(4.0, 4.0, 4.7) (2.0, 1.9, 1.6)	0.367
Using LLMs would improve my modeling performance	3.9 1.8	0.475	(4.3, 3.4) (1.8, 1.7)	**0.027***	(3.7, 3.5, 4.5) (2.0, 1.3, 1.8)	0.133
Using LLMs for modeling would increase my productivity	4.0 1.8	0.900	(4.2, 3.8) (1.9, 1.7)	0.303	(3.7, 4.0, 4.3) (1.9, 1.5, 2.0)	0.523
Using LLMs would enhance my modeling effectiveness	3.8 1.8	0.327	(4.0, 3.6) (1.8, 1.7)	0.324	(3.5, 3.8, 4.1) (2.0, 1.3, 1.8)	0.374
Using LLMs would make it easier to model	4.0 1.8	0.897	(4.4, 3.7) (1.9, 1.6)	**0.061***	(3.9, 4.0, 4.3) (1.8, 1.6, 2.0)	0.745
I would find LLMs useful for modeling	4.0 1.8	0.948	(4.4, 3.6) (1.8, 1.6)	**0.027***	(3.9, 3.8, 4.3) (1.8, 1.6, 1.9)	0.726
			Perceived Ease of Use			
Learning to operate LLMs would be easy for me	4.8 1.7	**<0.001***	(5.0, 4.5) (1.6, 1.7)	0.171	(4.5, 4.9, 5.0) (1.6, 1.5, 1.9)	0.249
I would find it easy to get LLMs to do what I want it to do	3.9 1.7	0.469	(4.1, 3.6) (1.8, 1.7)	0.260	(3.4, 3.8, 4.4) (1.7, 1.6, 1.8)	0.090
My interaction with LLMs would be clear and understandable	4.2 1.5	0.300	(4.5, 3.9) (1.5, 1.5)	0.070	(4.0, 3.9, 4.9) (1.6, 1.4, 1.6)	**0.022***
I would find LLMs to be flexible to interact with	3.9 1.6	0.621	(4.2, 3.6) (1.7, 1.5)	0.086	(3.6, 3.7, 4.6) (1.6, 1.3, 1.8)	**0.039***
It would be easy for me to become skillful at using LLMs	4.4 1.7	**0.051***	(4.9, 3.9) (1.5, 1.7)	**0.012***	(4.2, 4.2, 4.9) (1.8, 1.4, 1.8)	0.147
I would find LLMs easy to use	4.4 1.7	**0.167***	(4.5, 4.1) (1.7, 1.6)	0.217	(4.2, 3.9, 4.8) (1.5, 1.4, 1.9)	0.083

[1] (GPT-4, Llama), [2] (NPA, R4A, PTr), * Significant

answers, could easily go off topic, make up things;” (*ii*) Inaccuracy and unreliability – “The answers provided to me were not close to what I was looking for,” “it doesn’t understand complex questions,” “every time we tried to fix it or improve the diagram it made it worse;” (*iii*) Reliance on precise inputs – “Over caution, I kept fearing that my inputs needed to be extremely precise,” “need to explain everything like he is a little kid, every single aspect and point;” and (*iv*) Difficulty in visualization – “When asked to visualize/demonstrate something using drawing or some sort of UML diagram, it is not done in alignment and is somewhat messy, although there is potential and it can improve drastically.” This feedback may indicate the need for the development of concrete guidelines, such as prompt templates, for interacting with LLMs in conceptual modeling tasks. In our study, participants utilized entirely free-form text, which at times proved insufficient for eliciting the desired responses.

User Perception by Independent Variables (RQ2.2–RQ2.3). In Table 7, the results indicate a significant difference in perceived usefulness between GPT-

4 and Llama, in favor of GPT-4, and no significant differences in perceived ease of use. This latter result aligns with the fact that students were provided with the same interface and were unaware of the specific LLM they were using, thus not expected to encounter different challenges in usage. Nevertheless, we observe a significant difference in the assessment of ease of becoming skillful at using LLMs, favoring GPT-4. This can be explained by the more advanced capabilities of GPT-4, which may facilitate a better experience of use. The usefulness of the application, on the other hand, could depend on the utilized LLM. We see that the choice of LLM not only affects the overall perceived usefulness but also impacts perceived efficiency (quickness) and perceived effectiveness (performance). This suggests that more advanced LLMs like GPT-4 provide a better user experience by delivering faster and more accurate responses.

Table 7 further shows that, with respect to application domains, there are no significant differences in either perceived usefulness or ease of use. This may be attributed to the relatively low number of participants per application domain (ranging from 21 to 27). However, two significant differences were observed in terms of ease of use: understandability and flexibility. These results are surprising, considering that the application domains are comparable in terms of size and complexity, and LLMs are not biased toward any particular domain. Further research is required to explore the reasons for this result.

Summary of RQ2 Results. Based on the analyses conducted, we conclude that overall, the user experience was moderate (RQ2.1). The perceived usefulness of GPT-4 was found to be better than that of Llama, while there were no statistically significant differences in terms of perceived ease of use across LLMs (RQ2.2). Finally, there were no statistically significant differences in terms of perceived usefulness and perceived ease of use across application domains (RQ2.3).

5 Discussion

In the subsequent discussion, we outline the key practical implications derived from our results, along with an analysis of the potential threats to validity.

Practical Implications. Overall, the results suggest the potential of LLM-assisted conceptual modeling. However, the current way of interaction, relying solely on completely free-form text, presents limitations and can lead to frustration. Therefore, the first implication of our findings is the *development of prompt templates* for conceptual modeling, derived from the observed intention distribution. Expert modelers or domain experts can offer general, domain-, or language-specific templates for various modeling tasks. These templates should break down the overall task into smaller units, aiding modelers in clarifying their intentions and facilitating an iterative and incremental modeling process. As an example, consider the intention 'Discuss': "Discuss `<variants>` [in the setting of `<setting>`] [considering `<metric>` measures]," where `<variants>` specifies the variants to be discussed, `<setting>` is optional and specifies the setting in which the variants should be discussed, and `<metric>` suggests the metrics for assessing the variants, such as correctness or comprehensibility.

The second implication involves the *development of a recommender* to guide modelers regarding their next steps in the modeling process. This recommender would utilize our findings regarding the top recurrent interaction patterns observed in the study. By analyzing these patterns, the recommender could suggest relevant prompts or actions to users based on the current state of their modeling session. This approach aims to enhance the efficiency and effectiveness of the modeling process by providing modelers with tailored guidance. Moreover, by offering personalized recommendations based on the user's current modeling context and the identified interaction patterns, the recommender would seek to improve the user experience and perceived usefulness of LLM-assisted modeling.

The final implication underscores the importance of a *thorough tool selection*. With GPT-4 demonstrating superior perceived effectiveness and efficiency, it may significantly influence decisions regarding which LLM to employ for conceptual modeling. Further research needs to explore updated versions of these LLMs, alongside other alternatives. Moreover, analyzing the impact of prompt templates and recommenders on interaction behavior and user perception is crucial for a comprehensive understanding of LLM-assisted conceptual modeling.

Threats to Validity. In the discussion of the validity threats encountered in our study and the corresponding mitigation strategies, several key considerations emerged. To tackle concerns regarding *conclusion validity*, particularly concerning sample size, as our study encompassed 39 groups, we opted to assign each group two modeling tasks. This decision yielded a total of 566 interactions, thus enhancing the robustness of our conclusions. Concerning *construct validity*, particularly regarding the selection of intentions, we adopted a data-driven approach, carefully analyzing all prompts within the research team, with each prompt analyzed by two researchers to reconcile any inconsistencies and establish consensus. Addressing *internal validity*, the potential impact of variations in participants' capabilities and LLM assignment on the study's integrity was mitigated by randomly assigning tasks, ensuring equitable distribution, and minimizing bias. Furthermore, in addressing *external validity* regarding the generalizability of our findings across application domains, we selected three distinct domains familiar to both students and LLMs, thereby enhancing the potential transferability of our conclusions. Notably, we do not see a threat in having students participate in our experiments as our research focuses on intuitive LLM interactions, therefore there was no prerequisite to involve modeling experts [10].

6 Related Work

In recent studies, researchers have explored the application of LLMs in conceptual model generation. Chaaben et al. [6] demonstrated the effectiveness of few-shot prompt learning in completing static and dynamic domain models, emphasizing its versatility across various modeling activities. Giglou et al. [13] demonstrated the suitability of LLMs as assistants when fine-tuned for specific tasks. Chen et al. [8] conducted a comprehensive comparative study on using LLMs for fully automated domain modeling, revealing impressive domain understanding capabilities while emphasizing the need for careful consideration due to practical limitations. They observed that while LLMs provide reliable domain elements, there are often missing elements. Arul et al. [3] investigated how LLMs can be used to extract domain models from agile product backlogs. Kanuka et al. [17] explored the bidirectional traceability problem between design models and code, and they demonstrated the proficiency of ChatGPT in understanding and integrating specific requirements into design models and code. Ruan et al. [22] presented an automated framework for requirement model generation that incorporates ChatGPT-based zero-shot learning to extract requirement models from textual requirements and subsequently compose them using predefined rules. All these works concentrate on the LLM outcomes and their evaluation. Differently, our study concentrates on the human-LLM interaction and provides unique insights into interaction behavior and user perception.

Several works address prompt engineering, either generally or within the modeling context. White et al. [26] curated a catalog of prompt patterns that can be applied collaboratively throughout the software life-cycle, encompassing requirements elicitation, system design and simulation, code quality, and refactoring. In another paper, White et al. [25] suggest initial classifications for the catalog of prompt patterns tailored for use with ChatGPT, covering input semantics, output customization, error identification, prompt improvement, interaction, and context control. Fill et al. [12] explored how to generate and interpret ER, business process, UML class diagrams, and Heraklit models. Notably, these interactions were conducted by the researchers themselves with the LLM (GPT-4). While these works adopt a top-down approach for prompt catalog creation, we employed a data-driven approach that analyzes the intuitive interaction of humans with LLMs to extract interaction intentions. Moreover, we analyzed patterns of interactions rather than individual prompts.

7 Conclusion

In this paper, we conducted an empirical study to explore the interaction behavior and user perception of LLM-assisted conceptual modeling. Utilizing two LLMs (GPT-4 and Code-Llama), two modeling tasks (use case modeling with use case diagrams and domain modeling with class diagrams), and three application domains, we identified seven interaction intentions (create model, update model, create list, update list, present, explain and discuss) and five recurrent interaction patterns (model evolution, partial model visualization, partial model discussion, discussion-based model updating, and partial model listing).

In the future, we aim to expand our study by implementing a template-based approach designed to facilitate the interaction intentions and patterns identified in this research. Such an approach would offer a structured framework for users engaging with LLMs, enhancing efficiency and effectiveness in conceptual modeling. Additionally, we plan to broaden our investigation to encompass newer versions of various LLMs, allowing for a comprehensive assessment of their evolving capabilities. Finally, we will explore the impact of different interaction types on the quality metrics of the generated conceptual models, thereby providing deeper insights into enhancing LLM-assisted modeling processes.

References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Alex, N., Lifland, E., Tunstall, L., Thakur, A., Maham, P., Riedel, C.J., Hine, E., Ashurst, C., Sedille, P., Carlier, A., et al.: Raft: A real-world few-shot text classification benchmark. arXiv preprint arXiv:2109.14076 (2021)
3. Arulmohan, S., Meurs, M.J., Mosser, S.: Extracting domain models from textual requirements in the era of large language models. In: 2023 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C). pp. 580–587. IEEE (2023)
4. Brown, J.D.: The bonferroni adjustment. Statistics **12**(1), 23–27 (2008)
5. Cámara, J., Troya, J., Burgueño, L., Vallecillo, A.: On the assessment of generative ai in modeling tasks: an experience report with chatgpt and uml. Software and Systems Modeling **22**(3), 781–793 (2023)
6. Chaaben, M.B., Burgueño, L., Sahraoui, H.A.: Towards using few-shot prompt learning for automating model completion. In: 45th IEEE/ACM International Conference on Software Engineering: New Ideas and Emerging Results, NIER@ICSE. pp. 7–12. IEEE (2023)
7. Chen, B., Chen, K., Hassani, S., Yang, Y., Amyot, D., Lessard, L., Mussbacher, G., Sabetzadeh, M., Varró, D.: On the use of gpt-4 for creating goal models: an exploratory study. In: 2023 IEEE 31st International Requirements Engineering Conference Workshops (REW). pp. 262–271. IEEE (2023)
8. Chen, K., Yang, Y., Chen, B., López, J.A.H., Mussbacher, G., Varró, D.: Automated domain modeling with large language models: A comparative study. In: 2023 ACM/IEEE 26th International Conference on Model Driven Engineering Languages and Systems (MODELS). pp. 162–172. IEEE (2023)
9. Chen, X., Ye, J., Zu, C., Xu, N., Zheng, R., Peng, M., Zhou, J., Gui, T., Zhang, Q., Huang, X.: How robust is gpt-3.5 to predecessors? a comprehensive study on language understanding tasks. arXiv preprint arXiv:2303.00293 (2023)
10. Druckman, J.N., Kam, C.D.: Students as experimental participants. Cambridge handbook of experimental political science **1**, 41–57 (2011)
11. Du, X., Liu, M., Wang, K., Wang, H., Liu, J., Chen, Y., Feng, J., Sha, C., Peng, X., Lou, Y.: Evaluating large language models in class-level code generation. In: 2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE). pp. 865–865. IEEE Computer Society (2024)
12. Fill, H.G., Fettke, P., Köpke, J.: Conceptual modeling and large language models: impressions from first experiments with chatgpt. Enterprise Modelling and Information Systems Architectures (EMISAJ) **18**, 1–15 (2023)
13. Giglou, H.B., D'Souza, J., Auer, S.: Llms4ol: Large language models for ontology learning. In: The Semantic Web - ISWC 2023 - 22nd International Semantic Web Conference, Proceedings, Part I. Lecture Notes in Computer Science, vol. 14265, pp. 408–427. Springer (2023)
14. Grattafiori, W.X., Défossez, A., Copet, J., Azhar, F., Touvron, H., Martin, L., Usunier, N., Scialom, T., Synnaeve, G.: Code llama: Open foundation models for code. arXiv preprint arXiv:2308.12950 (2023)
15. Hou, Y., Zhang, J., Lin, Z., Lu, H., Xie, R., McAuley, J., Zhao, W.X.: Large language models are zero-shot rankers for recommender systems. In: European Conference on Information Retrieval. pp. 364–381. Springer (2024)

16. Izadi, M., Katzy, J., van Dam, T., Otten, M., Popescu, R.M., van Deursen, A.: Language models for code completion: A practical evaluation. arXiv preprint arXiv:2402.16197 (2024)
17. Kanuka, H., Koreki, G., Soga, R., Nishikawa, K.: Exploring the chatgpt approach for bidirectional traceability problem between design models and code. arXiv preprint arXiv:2309.14992 (2023)
18. Kocmi, T., Federmann, C.: Large language models are state-of-the-art evaluators of translation quality. arXiv preprint arXiv:2302.14520 (2023)
19. Liu, J., Liu, C., Lv, R., Zhou, K., Zhang, Y.: Is chatgpt a good recommender? a preliminary study. arXiv preprint arXiv:2304.10149 (2023)
20. MacFarland, T.W., Yates, J.M., MacFarland, T.W., Yates, J.M.: Kruskal–wallis h-test for oneway analysis of variance (anova) by ranks. Introduction to nonparametric statistics for the biological sciences using R pp. 177–211 (2016)
21. Qin, C., Zhang, A., Zhang, Z., Chen, J., Yasunaga, M., Yang, D.: Is chatgpt a general-purpose natural language processing task solver? arXiv preprint arXiv:2302.06476 (2023)
22. Ruan, K., Chen, X., Jin, Z.: Requirements modeling aided by chatgpt: An experience in embedded systems. In: 31st IEEE International Requirements Engineering Conference, RE 2023 - Workshops. pp. 170–177. IEEE (2023)
23. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
24. Van Veen, D., Van Uden, C., Blankemeier, L., Delbrouck, J.B., Aali, A., Bluethgen, C., Pareek, A., Polacin, M., Reis, E.P., Seehofnerova, A., et al.: Clinical text summarization: Adapting large language models can outperform human experts. Research Square (2023). https://doi.org/10.48550/ARXIV.2309.07430
25. White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., Schmidt, D.C.: A prompt pattern catalog to enhance prompt engineering with chatgpt. arXiv preprint arXiv:2302.11382 (2023)
26. White, J., Hays, S., Fu, Q., Spencer-Smith, J., Schmidt, D.C.: Chatgpt prompt patterns for improving code quality, refactoring, requirements elicitation, and software design. arXiv preprint arXiv:2303.07839 (2023)
27. Zhang, B., Haddow, B., Birch, A.: Prompting large language model for machine translation: A case study. In: International Conference on Machine Learning. pp. 41092–41110. PMLR (2023)
28. Zhang, T., Ladhak, F., Durmus, E., Liang, P., McKeown, K., Hashimoto, T.B.: Benchmarking large language models for news summarization. Transactions of the Association for Computational Linguistics **12**, 39–57 (2024)
29. Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al.: A survey of large language models. arXiv preprint arXiv:2303.18223 (2023)