On the Use of GPT-4 for Creating Goal Models: An Exploratory Study

Boqi Chen[*]
Electrical and Computer Engineering
McGill University
Montreal, Canada

Kua Chen[*]
Electrical and Computer Engineering
McGill University
Montreal, Canada

Shabnam Hassani[*]
School of EECS
University of Ottawa
Ottawa, Canada

Yujing Yang[*]
Electrical and Computer Engineering
McGill University
Montreal, Canada

Daniel Amyot
School of EECS
University of Ottawa
Ottawa, Canada

Lysanne Lessard
Telfer School of Management
University of Ottawa
Ottawa, Canada

Gunter Mussbacher
Electrical and Computer Engineering
McGill University
Montreal, Canada

Mehrdad Sabetzadeh
School of EECS
University of Ottawa
Ottawa, Canada

Dániel Varró
Linköping University
McGill University
Linköping, Sweden / Montreal, Canada

***Abstract*—The emergence of large language models and conversational front-ends such as ChatGPT is revolutionizing many software engineering activities. The extent to which such technologies can help with requirements engineering activities, especially the ones surrounding modeling, however, remains to be seen. This paper reports on early experimental results on the potential use of GPT-4 in the latter context, with a focus on the development of goal-oriented models. We first explore GPT-4's current knowledge and mastering of a specific modeling language, namely the Goal-oriented Requirement Language (GRL). We then use four combinations of prompts – with and without a proposed textual syntax, and with and without contextual domain knowledge – to guide the creation of GRL models for two case studies. The first case study focuses on a well-documented topic in the goal modeling community (Kids Help Phone), whereas the second one explores a context for which, to our knowledge, no public goal models currently exist (Social Housing). We explore the interactive construction of a goal model through specific follow-up prompts aimed to fix model issues and expand on the model content. Our results suggest that GPT-4 preserves considerable knowledge on goal modeling, and although many elements generated by GPT-4 are generic, reflecting what is already in the prompt, or even incorrect, there is value in getting exposed to the generated concepts, many of which being non-obvious to stakeholders outside the domain. Furthermore, aggregating results from multiple runs yields a far better outcome than from any individual run.**

***Index Terms*—Large Language Models, GPT-4, ChatGPT, Goal Modeling, Goal-oriented Requirement Language, GRL**

I. Introduction

The use of large language models (LLMs) for software engineering activities has recently caught the attention of the modeling community [1], [2], given the significant potential of this new technology. Today, one of the most powerful LLMs is OpenAI's GPT-4 [3] but it is not evident how much such an LLM can help with modeling activities, especially for early requirements. In this paper, we set out to examine the modeling knowledge of GPT-4 and its ability to create models through a series of exploratory experiments, focusing on goal modeling.

We identify a priori three challenges and two risks in using GPT-4 for goal modeling. The first challenge is the potentially incomplete knowledge of goal-oriented modeling languages in GPT-4, which is due to training needs. The second challenge has to do with random variation in the outputs from GPT-4. Thirdly, there is a need for "ground truth" or stakeholder input to assess the quality and validity of outputs. Meanwhile, the first risk is hallucination, i.e., GPT-4 may generate plausible-looking goal models containing important syntactic or semantic errors. Secondly, GPT-4 can generate highly generic outputs, i.e., models with obvious elements (actors, goals, etc.) that are not sufficiently specific to the domain to be useful. For example, some softgoals such as "usability" can be found for any system but do not allow for identifying any key conflicts and the need for trade-offs.

Motivated by these challenges and risks, we use the Goal-oriented Requirement Language (GRL) [4], and more specifically its textual syntax TGRL, in four experiments. In Section II, we first give a brief overview of GRL, LLMs, and related work. Section III presents the setup, results, and discussion for each of the experiments. The first experiment tests GPT-4 on baseline goal modeling knowledge (see Section III-B). The second experiment tests GPT-4 on the Kids Help Phone domain – a well-known case study in the goal modeling community [5] [6] (see Section III-C). The third

[*] All four authors contributed equally to this research.

Partially supported by NSERC Discovery/Discovery Accelerator programs, the FRQNT-B2X project (file number: 319955), IT30340 Mitacs Accelerate, and the Wallenberg AI, Aut. Systems & Software Program (WASP), Sweden

experiment tests GPT-4 on the domain of social housing, for which, to our knowledge, no public goal models currently exist (see Section III-D). The fourth experiment tests GPT-4's ability to improve a goal model interactively and incrementally through iteration and interaction (see Section III-E). Section III concludes with threats to validity before Section IV discusses the main findings of the experiments: (1) GPT-4 preserves considerable knowledge on goal modeling, (2) there is value in getting exposed to the ideas generated by GPT-4, (3) the responses have to be evaluated carefully as some are incorrect either syntactically or semantically, (4) many responses are generic and do not contribute much to the identification of conflicts among stakeholders, and (5) it is important to run GPT-4 multiple times (or in an interactive or incremental mode) as the aggregated results yield a much better set of goal model elements than any individual run. The conclusion in Section V summarizes the paper and discusses future work.

II. Background and Related Work

We first provide necessary background on GRL as well as on large language models before discussing related work.

The **Goal-oriented Requirement Language (GRL)** is a standardized goal modeling language that is part of the User Requirements Notation [4]. A GRL model is composed of actors that contain various types of intentional elements (i.e., goals, softgoals, tasks, resources, or beliefs) and indicators. These model elements can have an importance attribute (of intentional elements to their containing actor, and of actors to the model), labels, and textual descriptions. Various links exist in GRL, including contributions (with quantitative or qualitative levels), decompositions (AND, OR, XOR), and dependencies. GRL models enable capturing systems and their stakeholders, together with their objectives and quality requirements, in a way that enables rationale documentation as well as trade-off analysis and decision making. Standard GRL comes with a graphical notation as well as a textual notation (TGRL).

Recently, **large language models (LLMs)** have gained significant attention in natural language processing (NLP). LLMs are originally designed specifically for text generation. Given a sequence of tokens, e.g., words, $s = \{s_1, s_2 ..., s_{k-1}\}$, LLMs use deep neural networks, typically with the transformer architecture [7], to estimate the probability distribution of the next token $P(s_k|s_1, ..., s_{k-1})$.

Research has shown that, when trained on a sufficiently large corpus, an LLM has the capacity to preserve a substantial amount of knowledge implicitly within its parameters [8]. The resulting LLM can be queried for different kinds of knowledge and can answer questions in a domain without further fine-tuning or training but through prompt engineering [9], [10].

Related work covers work on the (semi-)automatic generation of goal models as well as the application of LLMs to model-driven engineering.

Güneş et al. [11] construct a goal model from a set of user stories by using NLP techniques to extract role names, actions, and benefits information from user stories, and then combine this information in different ways to build goal models.

While LLMs such as GPT [12] and BERT [13] are gaining growing interest, their adoption in model-based software engineering is still limited. Some approaches depend on fine-tuning a pre-trained LLM with a small dataset, such as using RoBERTA for meta-modeling [14]. More recently, due to the scarcity of task-specific datasets and generalizability of LLMs, more work has begun to investigate whether LLMs can be directly applied to tasks that can be reformulated as text generation, such as domain modeling [2], [15].

Zhou et al. [16] present an interactive and iterative modeling approach that merges human decision-making with deep learning, specifically BERT. This approach reduces goal modeling costs while maintaining model quality. Through interviews, the authors identified practical needs of goal modelers for automating modeling using the iStar goal modeling notation [17]. Based on these findings, the proposed hybrid approach combines deep learning-based entity and relational extraction with logical reasoning using dependency and statistical rules.

Wu et al. [18] propose an approach to generate iStar goal models [17] from user stories. The first step is node identification, where NLP techniques are used to extract 'who', 'what', and 'why' components from user stories. Node merging is then performed using BERT to calculate node embeddings. Pairs of nodes with a cosine similarity score above a certain threshold are merged. Finally, various kinds of relationships between nodes are identified based on predefined rules.

Fill et al. [10] demonstrate the utility of LLMs in conceptual modeling tasks. They experiment with GPT-4 to generate and interpret conceptual models, including Entity-Relationship diagrams, business process diagrams, UML class diagrams, and Heraklit models, underscoring the potential of LLMs in this domain.

White et al. [9] introduce prompt patterns to solve general problems in LLM interaction like ChatGPT. This research offers valuable insights into how prompting influences the performance of these models.

Compared to related work, our work is the first to apply GPT-4 for goal model generation. Furthermore, we perform a considerable amount of experimentation in order to evaluate the extent to which GPT-4 can answer questions on goal modeling concepts and generate goal models from descriptions in both stand-alone (i.e., GPT-4 gives a response to a single question) and interactive settings (i.e., GPT-4 responds to each question in a series of questions that build on each other).

III. Experimental Design and Results

This section assesses how much knowledge GPT-4 retains about goal modeling. We further evaluate how GPT-4 can use the retained knowledge to create goal models given a brief textual description. We also analyze the strengths and weaknesses of GPT-4 in goal modeling. Specifically, we design and conduct experiments to answer the following research questions (RQs):

TABLE I: Example Questions for each Category for Experiment B

Concept	Open	Explain the difference between a softgoal and a goal in GRL.
	Closed	What are all the types of qualitative contributions supported by GRL? Provide a one-sentence description for each of them.
Application	Open	Give me a sample goal model in the Goal-oriented Requirement Language (GRL), with 2 actors that have several goals each, as well as relationships.
	Closed	In GRL, indicators use target, threshold, and worst-case values as parameters to convert an evaluation value into a GRL satisfaction level (on a [0..100] scale). As they only have three parameters, such indicators are however limited in terms of functions they can capture. Sometimes, a complex function requires one to combine many partial indicators. Create a small GRL model (with one goal linked to as many indicators as you need) that determines whether a patient with diabetes has a satisfactory blood glucose level when not eating (to avoid hyperglycemia and hypoglycemia). - Hypoglycemia: Glucose level less than 0.60 g/l - Normal: Glucose level between 0.80 g/l and 1.00 g/l (ideally at 0.90 g/l) - Hyperglycemia: Glucose level above 1.10 g/l Describe your GRL model below. For each indicator, indicate its target, threshold, and worst-case values as well as its unit.

1) How much goal modeling knowledge does GPT-4 preserve?
2) How does GPT-4 perform in goal model generation from textual descriptions with different levels of detail?
3) How does immediate interactive feedback affect the quality of goal models generated by GPT-4?

A. *Experiment Setup*

a) Large language model: We use OpenAI's GPT-4 [3] as the LLM across multiple experiments. Specifically, the OpenAI GPT-4 API [19] is utilized for the independent prompts in research questions RQ1 and RQ2, while the interactive experiment for RQ3 uses the ChatGPT web interface with GPT-4 as the base model.

b) Dataset: For RQ1, we select 18 short-answer questions created by goal modeling experts. These questions are potential exam questions for undergraduate courses in requirements engineering. In RQ2 and RQ3, we select two GRL modeling questions that are also potential exam questions for the same courses. One modeling question focuses on a well-known domain in goal modeling literature, whereas the other modeling question uses a domain for which, to our knowledge, goal models are not publicly available.

More details about the dataset are provided in the respective subsection for each experiment. Furthermore, the prompts and grading results of all experiments can be found at: https://github.com/ChenKua/GRL_GPT

B. *Experiment B (Baseline Knowledge)*

a) Setup: The purpose of this experiment is to gauge baseline knowledge through direct questions related to constructs, syntax, and available tools. These questions assess the model's proficiency in handling queries ranging from simple closed questions to more complex, open-ended ones.

We organize these questions into 4 categories from two dimensions. The first dimension includes `Concept`, where questions focus on the meaning of one or more goal-modeling concepts, and `Application`, where the questions apply these concepts to solve some tasks. The second dimension considers whether the question is `Open` and thus has open-ended answers or `Closed` and thus has a unique answer. Table I illustrates all four categories along with one sample question for each category. Overall, we use the following categories: 5 `Open Application` questions, 3 `Open Concept` questions, 1 `Closed Application` question, and 9 `Closed Concept` questions. A balanced distribution over categories is a secondary concern to providing realistic exam questions during the selection of the questions for this experiment.

In the *first round (R1)*, we carry out the experiment including 18 prompts without providing any context. To account for potential random variation in responses from ChatGPT, each prompt is executed four times (*runs 1 to 4*) with independent API calls. The questions are posed to the model in an isolated manner, ensuring that each query is independent of the others. This setup allows us to evaluate the model's standalone reasoning capabilities and its ability to interpret and respond to questions based on its internal knowledge.

By using only the question text, there is the possibility that the model may misinterpret key modeling terms, e.g., GRL is interpreted as "Graph Representation Learning". To examine whether this is true, we conduct a second round (R2) of the experiment, this time providing a context prompt to each question as *"You are a software engineering student on an exam for goal-oriented requirement engineering. Follow the instructions in the question and answer the following question concisely."* With such a prompt, we also evaluate how well the model performs when it has a clear task context.

To conduct a systematic assessment of the model's performance, we adopt a grading schema involving four authors of this paper, who are graduate students from two universities. Each student independently grades the model's responses for the 18 questions, with two runs per round, i.e., each run is graded by two students. The score range for each question is an integer from 0 to 5, where 0 represents the answer is totally incorrect and 5 represents the answer is fully correct.

b) Results: Since each run is evaluated by two graders, we initially assess grading consistency. As the categories are ordinal (from 0 to 5), we use the Kendall rank correlation coefficient [20] to calculate the relationship between the scores assigned by the two graders. We get an average correlation coefficient value of 0.793, which corresponds to a strong relationship [21]. The detailed statistics of the Kendall rank correlation coefficient for each run are provided in Table II.

Table III shows the average scores for 18 questions per

TABLE II: Agreement Score: Kendall Rank Correlation for Different Runs for Experiment B

	Run 1	**Run 2**	**Run 3**	**Run 4**
R1	0.834	0.835	0.845	0.939
R2	0.886	0.592	0.532	0.884

TABLE III: Average Score of all Questions for Experiment B

	Run 1	**Run 2**	**Run 3**	**Run 4**	**SD**
R1	3.00	2.97	2.39	2.44	0.29
R2	3.39	3.75	3.78	3.86	0.18

round per run, along with a standard deviation (SD) between these average scores (last column). When comparing average scores between the two rounds, we observe a notable improvement in the average scores of all runs in R2 compared to R1. Furthermore, R2 exhibits a reduced standard deviation, implying that scores from R2 vary less than R1. This suggests that additional context prompts can generally enhance the performance of ChatGPT. We further calculate the average scores for each type of question and compare their difference between the two rounds.

Table IV provides the average and standard deviation scores for all four types of questions per round. Upon comparing the results in the two rounds, a significant increase in the average score is found for the `Concept` questions from R1 to R2, while the results of the `Application` questions remain relatively consistent. This finding suggests that the inclusion of context in the prompt is beneficial in scenarios where the context is unclear. For example, in some concept questions, GPT-4 misinterprets GRL as Graph Representation Learning or other technologies, rather than the Goal-oriented Requirement Language. An illustrative example of such a case can be seen in the first row of Table I: *"Explain the difference between a softgoal and a goal in GRL."*. In this case, GPT-4 incorrectly interprets GRL as Graph Representation Learning, resulting in an inaccurate response.

We also observe that `Closed Concept` questions (such as true or false conceptual questions) pose the most significant challenge to GPT-4 because the model knows part of the concepts but not all. Meanwhile, GPT-4 generally performs better on `Open` questions than on `Closed` questions. For the setting of `Open Application` questions, where GPT-4 is requested to provide a sample model without the specified domain, it produces excellent examples for most cases. However, when it comes to questions where reasoning capability is required (`Closed Application` questions), GPT-4 exhibits a substantial decrease in performance.

TABLE IV: Results of Two Rounds for Experiment B

R1	**Application**	**Concept**
Open	4.38 ± 0.26	3.25 ± 2.30
Closed	2.13	1.60 ± 1.48
R2	**Application**	**Concept**
Open	4.23 ± 0.46	4.75 ± 0.27
Closed	2.5	3.18 ± 1.72

c) Discussion: Overall, GPT-4 shows a certain level of understanding of concepts in goal modeling. Its average score for all 18 questions in the 4 runs in R2 is 3.68/5 (73.8%), which is equivalent to the letter grade B in university courses.

Moreover, the additional context prompt helps GPT-4 to understand domain-specific goal modeling terms correctly. Such a context prompt is especially important for questions with short descriptions where GPT-4 may not interpret correctly the context of the description. For questions with longer descriptions, the questions are detailed enough for GPT-4 to infer the context correctly.

We notice that GPT-4 performs much better in open questions compared to closed questions, which means it lacks some core goal modeling ability to create reliable goal models. Based on this observation, at inference time, practitioners can generally use GPT-4 to provide a high-level prototype for the models but should not expect GPT-4 to provide a complete model. Our next experiments investigate this idea in detail. However, when it comes to closed questions, our results suggest that practitioners should pay careful attention to the answers from GPT-4.

C. Experiment K (Kids Help Phone)

a) Setup: The purpose of this experiment is to evaluate how well GPT-4 can create from scratch a TGRL goal model for a well-known domain, given a prompt with varying degrees of context and formatting information. The domain selected for this experiment is the Kids Help Phone application [5] [6], which is well-studied in the goal modeling community. TGRL is chosen because a standardized textual grammar exists [4].

The content elements of the four prompts for the experiment are shown in Table V. The first prompt includes only a single sentence about the domain and mentions the three main stakeholders. The second prompt expands on the first prompt by adding a paragraph about the domain. This paragraph is taken verbatim from the literature [5]; we thus expect that GPT-4 should be aware of it as it should have appeared in the training data of GPT-4. The third and fourth prompts add a description of the TGRL syntax before the Kids Help Phone domain description in the first and second prompts, respectively. The syntax example is taken from the URN standard [4] and is from a different domain.

Each prompt is run four times to account for random variation in the responses of ChatGPT. A new session is started for the API call of each prompt. Each response is assessed by two authors based on the ground-truth goal model available in the same publication as the domain paragraph used for the second and fourth prompts [5]. Any disagreements between the assessments are discussed by the two authors to reach consensus. The assessment evaluates how much of the ground-truth goal model is covered by a response, how many elements in a response are not in the ground truth but are nonetheless reasonable, how many mistakes are made in a response, and how the responses differ depending on the prompt.

The ground-truth goal model contains 3 actors, 18 softgoals, 3 goals, 8 tasks, 2 resources, 38 contributions, and 12 depen-

TABLE V: Contents of Prompts for Experiment K. *Prompt 1*: Single Sentence; *Prompt 2*: Single Sentence + Domain Paragraph; *Prompt 3*: Syntax Description + Single Sentence; and *Prompt 4*: Syntax Description + Single Sentence + Domain Paragraph

Single Sentence	Using the textual grammar for the Goal-oriented Requirement Language (GRL), please provide a goal model for a Kids Help Phone application meant to provide online counselling for Canadian children including different actors such as the counsellors, the counselling organization, and youth and kids.
Domain Paragraph	Domain Context: The not-for-profit organization focuses on counseling for youth over the phone, but must now expand their ability to provide counseling via the Internet. Online counseling could be viewed by multiple individuals and may provide a comforting distance which would encourage youth to ask for help. However, in providing counseling online, counselors lose the cues they would gain through live conversation, such as timing or voice tone. Furthermore, there are concerns with confidentiality, protection from predators, public scrutiny over advice, and liability over misinterpreted guidance. The organization must choose among multiple technical options to expand their internet counseling service, including a modification of their existing anonymous question and answer system, discussion boards, wikis, text messaging, chat rooms. In order to make strategic decisions, a high-level understanding of the organization, system users, and the trade-offs among technical alternatives is needed.
Syntax Description	Assume a textual grammar called Goal-oriented Requirement Language (GRL) for modeling actors, their intentions, and their relationships. The language supports many types of intentions (goal, softgoal, task, indicator, belief, resource), one type of actor, and three types of relationships (dependsOn, contributesTo, decomposes). Actors and intentions may also each have an importance level (integer) and a description (string). Here is an example of the syntax: **actor** TelP#”Telecom Provider” { **importance** 100 **goal** VoiceConn#”Voice Connection Be Setup” { **importance** 50 } **softgoal** HighRel#”High Reliability” { **description** ”This is the most important objective of the stakeholder.” **importance** 75 } **softgoal** SpecUsage#**”Minimize Spectrum Usage”** { **importance** 60 } **task** MakeVoiceOverInternet#”Make Voice Connection Over Internet” { **contributesTo** HighRel **with** **somePositive** **contributesTo** SpecUsage **correlated** **with** **somePositive** **xor** **decomposes** VoiceConn } **task** MakeVoiceOverWireless#”Make Voice Connection Over Wireless” { contWirelessVoiceConnToHighRel **contributesTo** HighRel **with** **make** **contributesTo** SpecUsage **correlated** **with** **someNegative** **xor** **decomposes** VoiceConn } **indicator** VoiceConnFailureRate#”Failure Rate for Voice ConnectionOver Internet” { **unit** ”failures/week/10000 connections” contVoiceConnFailureRateToInternetVoiceConn **contributesTo** MakeVoiceOverInternet **with** 100 **dependsOn** Tech.LoggEquip } **belief** WirelessReliability#”Wireless is less reliable than Internet” { **contributesTo** HighRel **with** **SomeNegative** } } **actor** Tech#”Technician” { **resource** LoggEquip#”Logging Equipment” { **dependsOn** EquipSetup } **task** EquipSetup#”Correctly setup logging equipment” { **importance** 100 } }

dencies. The single sentence mentions all 3 actors and 2 goals, while the domain paragraph additionally covers 6 softgoals and 6 tasks. Each response element is evaluated based on four categories: correct (=1), partially correct (=0.5, i.e., a model element from the ground truth is covered but some mistake is made), incorrect, and reasonable (i.e., the model element could reasonably be in the ground truth but is not).

b) Results: Table VI shows the average percentage of intentional elements from the ground truth covered by the responses for each prompt. The sentence column shows the covered elements that also appear in the single sentence. The paragraph column shows the covered elements that also appear in the domain paragraph but not in the single sentence. The fourth column shows the covered elements that are not in the prompt. Furthermore, all three actors are covered by each response regardless of the prompt (100%), and the average percentages of relationships covered by the responses are negligible for all prompts, ranging from 0% to 4% only.

TABLE VI: Average Percentage of Intentional Elements from Ground Truth in Responses for Experiment K

Prompt	Sentence	Paragraph	Not in Prompt	Total
1	2.0	8.1	8.1	18.1
2	3.2	9.7	3.2	16.1
3	4.4	1.2	2.4	8.1
4	5.6	8.1	4.0	17.7

Table VI shows that the overall results are similar for all prompts, except for the third prompt (i.e., single sentence + syntax description). It seems that the long syntax description overshadows the single sentence about the domain. In general, a longer domain description does not mean a better response because the paragraph and total results for prompt 1 (shorter domain description) are similar to the results of prompts 2 and 4 (longer domain description). Note that the paragraph column for prompts 1 and 3 means that concepts from the domain paragraph are covered by the response even though they are not in the prompt, whereas the same column for prompts 2 and 4 means that these covered concepts are in the prompt.

Table VII shows the average percentage of intentional elements from the single sentence and domain paragraph that are covered by the responses for each prompt. The paragraph column includes only elements that are not already in the single sentence. Furthermore, the three covered actors all appear in the single sentence, and no relationships from the ground truth appear in any prompts.

TABLE VII: Average Percentage of Intentional Elements from Prompt in Responses for Experiment K

Prompt	Sentence	Paragraph
1	31.3	20.8
2	50.0	25.0
3	68.8	3.1
4	75.0	20.8

Overall, the responses do not cover well the intentional elements mentioned in the prompts with at most 25% (i.e., 4 out of 12) of the elements from the domain paragraph being covered. There is no clear difference between prompts 1 and 3 (shorter domain description) and prompts 2 and 4 (longer domain description).

Finally, the second and third columns in Table VIII show the average percentage of incorrect and reasonable elements (actors, intentional elements, and relationships combined) out of all response elements, respectively. The fourth column shows the total average number of response elements.

TABLE VIII: Average Percentage of Elements in Responses and Total Number of Elements for Experiment K

Prompt	Incorrect	Reasonable	Total
1	19.7	47.7	35.5
2	15.5	49.8	37.8
3	5.1	77	34.0
4	10.8	56.9	26.8

The results clearly show that a large part of the responses is reasonable but there are also significant mistakes in the responses. Prompts 3 and 4 with the syntax description result in fewer mistakes on average and a larger number of response elements on average that are not in the ground truth but are still reasonable. The average size of the response (i.e., the total number of response elements) is also similar across all prompts except for prompt 4 where the response is shorter on average than for the other prompts.

c) Discussion: The responses provided by GPT-4 exhibit a large variation. For example, the total number of response elements ranges from around 25 to around 50 for each of the prompts except for the fourth one which is more uniform with around 25 response elements. Furthermore, the type of model elements covered by a response differs from one response to another. All responses cover actors, goals, and tasks. Most cover softgoals but only some cover resources, beliefs, or indicators. The variation is highest for relationships, with some responses containing no meaningful relationships at all while other responses cover contributions, decompositions, and dependencies. The responses to prompts with the syntax description almost always cover all types of relationships.

Although we cannot substantiate it quantitatively, the observation of the authors who assessed the responses is that understandability of a response varies a lot, and again, the responses to the prompts with syntax description are much more readable. Certainly, these responses largely follow the provided syntax. Similarly, the softgoal/goal/task hierarchy is modeled quite well in the responses to prompts with syntax description, while other prompts often result in no attempt to model the hierarchy.

Finally, the large number of incorrect response elements is somewhat misleading because, often, a single type of mistake is responsible for all incorrect elements. Responses to prompts without syntax description often use the wrong type for an intentional element (typically goal instead of softgoal), do not assign an intentional element to an actor, or have wrong relationships (e.g., without a source element or two types of relationships between the same elements). In contrast, responses to prompts with syntax descriptions rarely make mistakes with the type of intentional elements. The most prominent mistake for those responses is to place relationships outside intentional elements in a separate section (i.e., a syntax mistake rather than a semantic mistake). There are also mistakes related to using an XOR instead of an AND decomposition as well as using dependencies within an actor. However, some mistakes can be attributed to the fact that the syntax description in the prompt is limited and does not cover all cases. In general, there are only a few cases where the response does not make sense regardless of which prompt is used.

Overall, all responses on average, regardless of the prompt, definitely receive a passing grade for a university exam exercise that does not require deep domain knowledge. There are even cases for each prompt where there are only one or two mistakes, and two responses to the third prompt have not a single mistake, which would deserve a decidedly higher grade. However, the responses can be somewhat generic as up to approximately 60% of the reasonable response elements (i.e., not in the ground truth but could be) that are also not mentioned in the prompt can be rather domain-independent. Note that it is difficult to decide which response is generic and which one is not, so these results need to be interpreted with caution. This also means that at least 40% of the reasonable response elements that are not mentioned in the prompt do contain valuable information. On average, there are between 1 and 7 domain-specific response elements per run that represent new information not available in the prompt. This large variation leads to the conclusion that the best strategy is to repeat the same prompt a number of times to maximize exposure to new ideas in the responses.

D. Experiment S (Social Housing)

a) Setup: The purpose of this experiment is to evaluate how well GPT-4 can create a TGRL goal model of a little-known domain. It follows the process of Experiment K in terms of the number and types of prompts. The chosen domain is Social Housing, which, to date, has been the subject of limited research by the requirements engineering community. Some research has focused on requirements for social housing projects [22], but none, to our knowledge, has led to publicly available documentation regarding the use of goal modeling for this domain. Two of the authors of this paper are however actively involved in an ongoing project aiming to develop a dashboard for social housing planning and management, and lead the research team responsible for the development of a prototype dashboard.

The three content elements for the prompt are similar in style and length to those used for Experiment K. There is thus a first content element containing a single sentence, a second one containing a paragraph about the domain (see Table IX), and a third using the same description of TGRL as used for Experiment K (last row of Table V). These elements are used in the same manner as for Experiment K, generating: 1) a short prompt with one sentence; 2) a longer prompt with the short sentence and the domain description; 3) the short prompt preceded by the syntax description; and 4) the longer prompt also preceded by the syntax description. Similarly, a new session is started for the API call of each prompt, and each prompt is run four times and assessed by the two authors with domain knowledge. Disagreements in the assessments are resolved by the two authors through discussion.

The quantitative assessment of Experiment S is more limited than for Experiment K due to a lack of recognized ground truth, but follows the same evaluation categories as Experiment

TABLE IX: Content Elements for Prompts for Experiment S

Single Sentence	Using the textual grammar for the Goal-oriented Requirement Language (GRL), please provide a goal model for a social housing application meant to support business intelligence and decision making for different actors such as the City of Ottawa, shelters, and the federal and provincial governments.
Domain Paragraph	Domain Context: To improve social housing planning and management in Canadian cities and regions, a social housing application is required. The application will integrate anonymized data collected in current social housing databases and be supported by a data warehouse with predictive capabilities. This solution should enable better decision-making related to the future development of housing stocks. However, the stakeholders that would use the application, including housing providers, government agencies and social housing applicants, have different and potentially conflicting roles and needs in terms of access to information, transparency, privacy, and granularity of predictions. Moreover, the format and quality of data in existing databases may limit the ability to run certain queries or hinder the quality of the results. Developing the application from scratch will require prioritizing among stakeholder goals and concerns and making trade-offs between technical capabilities and feasibility.

K. A simple goal model containing only actors and goals had been created by the Social Housing research team based on stakeholder interviews and documentation, and is used as a partial ground truth in this experiment. It contains 9 actors and 34 goals. The single sentence prompt mentions 4 of these actors and 2 goals. The domain paragraph provides 2 additional actors and 3 additional goals. It also mentions 5 softgoals, 3 tasks, 3 resources, and 1 negative contribution.

b) Results: Overall, the results for Experiment S are similar to but weaker than those of Experiment K. Answers to all prompts contain the 4 actors from the ground truth provided in the sentence prompts, with only a few answers containing one additional actor when prompted with the paragraph. The answers find 3% or less of the ground truth goals that are contained within the prompts, but up to 8% of the ground truth goals that are not mentioned in the prompts. Experiment K, on the other hand, finds up to approximately 18% of intentional elements from the ground truth (see Table VII).

TABLE X: Average Percentage of Elements in Responses and Total Number of Elements for Experiment S

Prompt	Incorrect	Reasonable	Total
1	24.2	59.9	37.8
2	21.8	55.0	29.9
3	24.2	56.5	25.5
4	18.6	36.7	30.5

GPT-4 fares better when also considering actors and other intentional elements that are reasonable for the domain of Social Housing without being present in the ground truth used for the experiment. Table X shows the average percentage of response elements that are reasonable vs. incorrect for each prompt. The last column shows the average number of elements given in response to each prompt, including those that are in the ground truth and those that are not. Overall, Table X shows that GPT-4 is able to find useful elements about a domain. The much lower percentage of reasonable elements identified when prompted with what should be the best prompt (sentence, paragraph, and syntax) is nevertheless surprising. Moreover, the reasonable elements identified across prompts are often taken from the prompts themselves. Considering only the reasonable elements that are not mentioned in the prompt, up to approximately 70% are generic for the prompts without syntax description and up approximately 35% are generic for the prompts with syntax description.

c) Discussion: Similarly to Experiment K, a large variation exists across answers to the same prompt in terms of the number and types of elements. Also, the answers provided to prompts 3 and 4, which contain the TGRL syntax description, are generally better. For example, up to approximately 65% of the response elements for prompts 3 and 4 are providing useful ideas that are not in the prompt compared to up to approximately 30% for prompts 1 and 2. The authors surmise that providing a syntax helps GPT-4 to know what to look for in the domain. Given the variation, however, the conclusion is again that several runs are needed to get an aggregated response that is far better than any individual response.

It is interesting to note that the overall size of the response is similar for Experiments K and S (see total number of elements in Tables VIII and X), even though the social housing domain is more complex than the Kids Help Phone domain. While the number of incorrect elements is higher for the social housing domain (which may be explained by stricter grading of this experiment), the number of reasonable elements is similar. Furthermore, the number of useful (i.e., domain-specific) elements in the reasonable elements that are also not in the prompt are comparable between Experiments K and S (at least 40% for the former and up to 65% for the latter), showing that there is overall useful information in the responses of GPT-4.

Similar to Experiment K, GPT-4 receives a passing grade for the social housing exercise. However, the results are generally worse than for Experiment K, which may be attributed to the more in-depth domain knowledge required for social housing.

E. Experiment I (Interactive)

a) Setup: The purpose of this experiment is to explore GPT-4's ability to improve its proposed TGRL goal model through iteration, thus following multiple prompts during the same session. This experiment centers on the Social Housing domain since the results of Experiment S have shown that GPT-4 does not perform very well in terms of identifying domain-specific intentional elements that are not already present in the prompts. Since these results could be related to the quality of the prompts rather than to GPT-4's capabilities, it is important to investigate if GPT-4's performance improves through the use of additional prompts.

The strategy for the interactive session is to start with a pre-existing prompt from the previous experiments, then to provide additional prompts to assess GPT-4's ability to correct errors in the models, and finally to provide prompts aiming to assess GPT-4's ability to include additional domain-specific elements in its model. All but one author attended the interactive session, and the follow-up prompts for GPT-4 were agreed upon collectively after discussing GPT-4's responses. The experiment as conducted is composed of 16 prompts and responses. The initial prompt is thus taken from the most comprehensive set of prompt elements from Experiment S (the TGRL syntax description, a sentence summarizing the problem space, and a paragraph providing a domain description). The follow-up prompts fall under the following categories:

- Correction of model syntax (e.g., "Please regenerate that model while specifying the type of decomposition (AND, OR, XOR), as indicated by the syntax.")
- Correction of model semantics (e.g., "Task QueryData does not contribute in any way to data quality. Please adjust the model accordingly.")
- Expansion of model (e.g., "I think other stakeholders may want to use that system. Please adjust the model accordingly.")
- Justification of model elements (e.g., "Why would potential tenants want to access the application?")

b) Results: The initial model is reasonable, identifying additional elements to those provided in the prompt, e.g., identification of an "Application Developers" actor. As with previous experiments, goals are rather generic and the model contains some syntactic and semantic errors.

GPT-4 is able to correct the syntax of its proposed TGRL models when provided with prompts with direct instructions. Its ability to correct the semantics of the model is more limited in the sense that it will modify the model to comply with the prompt, but without consideration for the relevance of intentional elements. For example, in response to the prompt "The provincial government is not responsible for managing housing stocks. Please adjust the model.", GPT-4 simply removes the element "Manage Housing Stock" from the model instead of attributing it to another actor.

Moreover, GPT-4 is unable to correctly modify the model in response to more complex prompts. For example, when told that housing stocks is a shared responsibility between the City of Ottawa and Housing Providers, GPT-4 responds that it is indeed a shared goal, but does not attribute the goal to both actors or use a dependency element between such a goal and these actors to express shared responsibility.

Also, syntactic corrections can lead to semantic modifications that are not necessarily correct. For example, when asked to ensure that the sum of the importance values of the intentional elements be 100, GPT-4 complies by attributing values in decreasing importance to intentional elements according to their position (highest value to first element), without consideration for the correctness of these values.

GPT-4 performs well when provided with open-ended prompts asking it to expand the model. For example, when asked to identify additional stakeholders who may want to use the social housing system, GPT-4 correctly identified potential tenants and social workers and provided a reasonable rationale for adding them to the model.

However, it does not update the model with elements that could indicate relationships among previously-identified and newly-identified actors (e.g., dependencies, contributions). This is a general issue, with the generated TGRL models containing few elements indicating relationships among actors.

When asked to identify conflicting goals, GPT-4 identifies common-sense conflicts (e.g., data availability vs. data privacy) in its response prior to the model, but incorrectly attributes these conflicts to specific actors. Also, it only partially expresses these conflicts as negative contributions in the model.

The TGRL models generated by GPT-4 tend to identify high-level goals of social actors, rather than goals that are directly related to their use of a system. When asked to provide system-relevant goals, GPT-4 mostly transformed social actors' goals into tasks when these goals had relational elements. The result was overall a much poorer model. When asked to forget that prompt and to provide the model that had been generated previously, GPT-4 rather generated a TGRL model that resembled the original one.

c) Discussion: Additional prompts can improve the syntax and semantics of TGRL models. However, intentional elements beyond what is required are often added, removed, or modified when regenerating the model. These unrequested modifications to the model can generate additional syntactic and semantic errors. Pre-existing domain expertise is thus required to create prompts related to correcting the meaning of the model's elements, and to assess the responses' validity.

Additional prompts within a session are useful for expanding an initial model within the domain, thus to identify additional actors and goals that are reasonable for that domain. However, GPT-4 fares poorly in identifying and correctly attributing relational elements among actors (e.g., contributions, typed decompositions, and dependencies). Moreover, GPT-4 is able to identify high-level, general goals of social actors, but has very limited capabilities in identifying goals that are relevant for the system to be created. As a result, the models can serve as an acceptable starting point to describe a domain but would be of limited use for analysis purposes.

TGRL models generated by GPT-4 appear to be path-dependent, thus earlier prompts within a session strongly influence the results of subsequent prompts without the possibility to truly "go back" to a previous point in the interaction (i.e., "undo" commands are not very effective).

As the starting point for Experiment I is a result from Experiment S for which GPT-4 already received a passing grade, GPT-4 also receives at least a passing grade for the interactive exercise. While the interactive mode can bring forward additional ideas, the syntactic and semantic quality of the responses cannot be taken for granted.

As a side note, GPT-4 gets an A+ for social skills, apologizing when prompts point out mistakes or supporting the prompt's statement: "You're right!", "Absolutely!". However, it does get lazy over time, using "..." to indicate repeated model elements when regenerating the model after multiple prompts.

F. Threats to validity

In this section, we discuss the internal, external, and construct validity of the four experiments.

a) Internal validity: To address potential bias in manual evaluation, especially when it comes to open-ended questions, we involve two graduate students in the evaluation process for each run for Experiment B. These students independently evaluate the generated answers and are required to provide comments along with their assigned scores, ensuring a thorough review process. We report the agreement score (Kendall rank correlation), which corresponds to a strong relationship.

For Experiments K and S, four authors with significant goal modeling expertise assess the generated goal models. Each goal model is assessed by two authors and agreement is reached on each response element's assessment. We acknowledge that we do not systematically evaluate the understandability of the responses for Experiments K and S but still report our observations as we find them useful. Similarly, how generic a response element is in the context of a domain can be subjective and rather difficult to agree on; we hence acknowledge this threat to validity.

For Experiment I, all but one author collectively assess the response of GPT-4 in the interactive (and hence unpredictable) setting and collectively decide on the next prompts to avoid individual bias.

b) External validity: A threat to the generalization of the results is the variations in the response which can be expected from LLMs. In our case, the response of GPT-4 exhibits slight variations with each run for Experiment B and more significant variations for Experiments K and S. To account for this variability, we perform each experiment four times and calculate the average score for all experiments and, in addition, the standard deviation for Experiment B. However, a larger number of runs would allow us to be more confident in our findings. In general, the low number of goal modeling exercises (Kids Help Phone and Social Housing) may also mean that results may not generalize to other domains.

c) Construct validity: The configuration of GPT-4 as well as the influence of earlier interactions on later interactions may influence the outcome of our experiments. We use a fixed (default) setting of 0.5 for GPT-4's level of randomness. Other settings may result in a different degree of variation in the responses. Furthermore, we ensure that the individual prompts are independent of each other for Experiments B, K, and S by starting a new session for each prompt.

IV. Discussion

Although GPT-4 is far from a perfect goal modeler, there is still value in getting exposed to the ideas generated by GPT-4. Meanwhile, the responses have to be evaluated carefully as some are incorrect either syntactically or semantically. We also notice that many responses are too generic to contribute much to the identification of conflicts among stakeholders. Another weakness is that we observe that GPT-4 has a limited reasoning ability. Specifically, it is able to answer prompt-based questions but suffers from a limited ability to synthesize information or make inferences from the knowledge base. One way to improve the responses from GPT-4 is to run it several times as the aggregated results yield a much better result than any individual run.

The difference in the results for Experiment K and S, i.e., the latter performed somewhat weaker than the former with fewer intentional elements being discovered, aligns with our initial intuition that an unknown domain in terms of availability of goal models such as social housing makes it more difficult for GPT-4 to respond appropriately. This is interesting as one would assume that there is ample information about social housing available online (and hence available to GPT-4). It seems that GPT-4 requires direct knowledge in a similar context. For the Kids Help Phone domain, literature at the intersection of goal models and the application domain exists. If such knowledge does not exist, then it seems that GPT-4 has issues connecting the dots.

Based on our experiments and findings, we answers the three research questions (RQs) as follows:

Answer to RQ1: How much goal modeling knowledge does GPT-4 preserve? We find that GPT-4 preserves considerable knowledge on goal modeling. It is able to achieve a letter grade of B in answering university-level exam questions on goal modeling.

Answer to RQ2: How does GPT-4 perform in goal model generation from textual descriptions with different levels of detail? GPT-4 exposes the modeler to useful ideas that may be non-obvious to stakeholders outside the domain. While it is valuable to include syntax information in the prompt, the amount of domain information has a limited effect on the responses of GPT-4. The responses have to be evaluated carefully as many elements generated by GPT-4 may be either incorrect or rather generic and hence not very conducive to highlight important conflicts among stakeholders in the domain. Aggregating results from multiple runs yields a far better outcome than from any individual run.

Answer to RQ3: How does immediate interactive feedback affect the quality of goal models generated by GPT-4? We find that immediate interactive feedback can improve the syntax and semantics of the TGRL model and expand the initial model for simple requests. However, it may cause unintended side-effects, and undoing a previous prompt may introduce model errors.

V. Conclusion

In this paper, we evaluate how GPT-4, a popular LLM, can perform goal-modeling tasks. Specifically, we use the GPT-4 API for three experiments and ChatGPT with GPT-4 as the backend LLM for the fourth experiment. Our evaluation contains three parts. In the first experiment (Experiment B – Baseline Knowledge), we evaluate how much background knowledge does GPT-4 retain about goal modeling. In the second experiment (Experiment K – Kids Help Phone and Experiment S – Social Housing), we test whether GPT-4 can generate an entire goal model in TGRL format with different levels of information provided. Finally, in Experiment I – Interactive, we evaluate how immediate feedback to GPT-4 (via ChatGPT) can affect the quality of the output models.

In all experiments, we find that GPT-4 retains certain levels of background knowledge about goal modeling, which is equivalent to at least a passing grade for a student in a university-level course. Moreover, in model generation tasks, we found that the GPT-4 output has a large variation, although syntax information in the prompt helps it to produce models with better quality. It is highly advisable to run the same prompt several times, as an aggregate result yields a much better set of goal model elements than any individual run. Due to variation in output, an individual run may result in a failing grade. Finally, in the interactive experiment, we find that GPT-4 can correct itself with intermediate feedback, however, it fails to react to feedback in more sophisticated cases.

We believe that our work provides an initial guide for using LLMs in goal modeling. We identify the strengths and limitations of an LLM for goal modeling in different scenarios. As future work, we plan to (1) expand on the experiments presented in this paper in terms of number of exercises and the size of goal models. Scalability needs to be investigated further, especially since the response size is the same for our two domains in experiments K and S, which are quite different in terms of complexity. We further plan to (2) investigate the impact of other prompts on the quality of the output. For example, the detail of the context information provided in a prompt could be varied based on the kind of question such as open vs closed question. We also plan to (3) investigate strategies for interacting with LLMs in terms of which directions to take when incrementally improving an initial output. For example, is it better to first ask an LLM to expand on the initial output with additional actors and intentional elements, or to correct existing syntactical mistakes, or to correct existing semantical mistakes? Finally, we plan to (4) investigate industrial guidelines and strategies in terms of how to deal with the high level of variation in the output of LLMs. For example, how often do responses need to be generated to yield a sufficient overall result, how to review each response and reach consensus, and how to combine individual responses into an overall result.

References

[1] B. Combemale, J. Gray, and B. Rumpe, "ChatGPT in software modeling," *Software and Systems Modeling*, vol. 22, pp. 777–779, 2023. [Online]. Available: https://doi.org/10.1007/s10270-023-01106-4

[2] J. Cámara, J. Troya, L. Burgueño, and A. Vallecillo, "On the assessment of generative AI in modeling tasks: an experience report with ChatGPT and UML," *Software and Systems Modeling*, vol. 22, pp. 781–793, 2023. [Online]. Available: https://doi.org/10.1007/s10270-023-01105-5

[3] OpenAI, "GPT-4 technical report," 2023. [Online]. Available: https://arxiv.org/abs/2303.08774

[4] ITU-T, "Recommendation Z.151: User Requirements Notation (URN) – Language definition," 2018. [Online]. Available: http://www.itu.int/rec/T-REC-Z.151/en

[5] J. Horkoff and E. Yu, "Interactive goal model analysis for early requirements engineering," *Requir. Eng.*, vol. 21, no. 1, p. 29–61, March 2016. [Online]. Available: https://doi.org/10.1007/s00766-014-0209-8

[6] J. Horkoff, "Using i* models for evaluation," Master's thesis, University of Toronto, Canada, 2006. [Online]. Available: https://www.cs.utoronto.ca/%7Ejenhork/MScThesis/Thesis.pdf

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[8] B. AlKhamissi, M. Li, A. Celikyilmaz, M. Diab, and M. Ghazvininejad, "A review on language models as knowledge bases," *arXiv preprint arXiv:2204.06031*, 2022.

[9] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. Elnashar, J. Spencer-Smith, and D. C. Schmidt, "A prompt pattern catalog to enhance prompt engineering with ChatGPT," *arXiv preprint arXiv:2302.11382*, 2023.

[10] H.-G. Fill, P. Fettke, and J. Köpke, "Conceptual modeling and large language models: impressions from first experiments with ChatGPT," *Enterprise Modelling and Information Systems Architectures (EMISAJ)*, vol. 18, pp. 1–15, 2023. [Online]. Available: https://doi.org/10.18417/emisa.18.3

[11] T. Güneş and F. B. Aydemir, "Automated goal model extraction from user stories using NLP," in *2020 IEEE 28th International Requirements Engineering Conference (RE)*. IEEE, 2020, pp. 382–387. [Online]. Available: https://doi.org/10.1109/RE48521.2020.00052

[12] T. Brown, B. Mann, N. Ryder *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, vol. 1, 2019, pp. 4171–4186.

[14] M. Weyssow, H. Sahraoui, and E. Syriani, "Recommending metamodel concepts during modeling activities with pre-trained language models," *Software and Systems Modeling*, vol. 21, no. 3, pp. 1071–1089, 2022.

[15] M. B. Chaaben, L. Burgueño, and H. Sahraoui, "Towards using few-shot prompt learning for automating model completion," in *2023 IEEE/ACM 45th International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*, 2023, pp. 7–12.

[16] Q. Zhou, T. Li, and Y. Wang, "Assisting in requirements goal modeling: a hybrid approach based on machine learning and logical reasoning," in *Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems*, 2022, pp. 199–209. [Online]. Available: https://doi.org/10.1145/3550355.3552415

[17] E. S. Yu, "Modeling strategic relationships for process reengineering," in *Social Modeling for Requirements Engineering*, 2011, pp. 66–87.

[18] C. Wu, C. Wang, T. Li, and Y. Zhai, "A node-merging based approach for generating iStar models from user stories," *Software Engineering and Knowledge Engineering*, pp. 257–262, 2022. [Online]. Available: https://people.cs.pitt.edu/~chang/seke/seke22paper/paper176.pdf

[19] OpenAI, "GPT-3.5 models." [Online]. Available: https://platform.openai.com/docs/models/gpt-3-5

[20] M. G. Kendall, "A new measure of rank correlation," *Biometrika*, vol. 30, no. 1/2, pp. 81–93, 1938.

[21] G. W. Corder and D. I. Foreman, *Nonparametric Statistics for Non-Statisticians*. John Wiley & Sons, Inc., 2011.

[22] J. P. Baldauf, C. T. Formoso, P. Tzortzopoulos, L. I. G. Miron, and J. Soliman-Junior, "Using building information modelling to manage client requirements in social housing projects," *Sustainability*, vol. 12, no. 7, 2020. [Online]. Available: https://www.mdpi.com/2071-1050/12/7/2804