[BKK+24] L. Burgueño, M. Keet, J. Kienzle, J. Michael, Ö. Babur:
A Human Behavior Exploration Approach Using LLMs for Cyber-Physical Systems.
In: Workshop on Artificial Intelligence and Model-driven Engineering, MODELS Companion '24:
International Conference on Model Driven Engineering Languages and Systems (MDE Intelligence),
pp. 578-586, Association for Computing Machinery (ACM), Oct. 2024.

A Human Behavior Exploration Approach Using LLMs for Cyber-Physical Systems

Lola Burgueño
lolaburgueno@uma.es
ITIS Software, University of Malaga
Spain

C. Maria Keet
mkeet@cs.uct.ac.za
University of Cape Town
South Africa

Jörg Kienzle
Joerg.Kienzle@uma.es
ITIS Software, University of Málaga
Spain

Judith Michael
michael@se-rwth.de
RWTH Aachen University
Germany

Önder Babur
onder.babur@wur.nl
Wageningen University & Research
Eindhoven University of Technology
The Netherlands

ABSTRACT

In the early phases of Cyber-Physical Systems (CPS) development, scoping human behavior plays a significant role, especially when interactions extend beyond expected behavior. Here, it is especially challenging to develop cases that capture the full spectrum of human behavior. Up to now, identifying such behavior of humans remains a task for domain experts. We explore how one can use Large Languages Models (LLMs) in the design phase of systems to provide additional information about human-CPS interaction. Our approach proposes a preliminary ontology describing a hierarchy of types of behavior and relevant CPS components as input for prompt templates. It uses them to generate parts of human behavior descriptions, as well as a canned prompt with one variable about behavior. For demonstration, we take a smart building with a Home Energy System as the use case.

An initial user evaluation shows that the behavior descriptions generated with standard and ontology-driven prompts complement each other and are useful when assisting humans. The discovered uncommon behaviors can be used to complete interaction scenarios that eventually result in a more robust CPS implementation.

CCS CONCEPTS

• **Software and its engineering** → **Designing software**; *Requirements analysis*; • **Information systems** → **Language models**; • **Computer systems organization** → *Embedded and cyber-physical systems.*

KEYWORDS

Human Behavior, Large Language Models, Cyber-Physical Systems, User Scenario, Digital Twin

MODELS Companion '24, September 22–27, 2024, Linz, Austria

ACM ISBN 979-8-4007-0622-6/24/09
https://doi.org/10.1145/3652620.3687806

ACM Reference Format:
Lola Burgueño, C. Maria Keet, Jörg Kienzle, Judith Michael, and Önder Babur. 2024. A Human Behavior Exploration Approach Using LLMs for Cyber-Physical Systems. In *ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems (MODELS Companion '24), September 22–27, 2024, Linz, Austria.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3652620.3687806

1 INTRODUCTION

The applications of Cyber-Physical Systems (CPS) span many domains, such as healthcare [42], transportation [37], buildings [20], and maritime [10]. These systems interact with the physical world through sensors, make environmental changes with actuators, and adapt themselves to evolving environmental conditions. In many of these systems, humans are part of the environment, thereby interacting with the CPS. CPSs face unpredictable behaviors due to many uncertain environmental conditions, including how humans interact with them. Not all possible behaviors (including normal and exceptional) are known during the CPS development due to limited knowledge about the CPS operating environment. Therefore, there is a great need to identify such behaviors so that the implementation of CPSs can be improved to deal with them, avoiding possible unsafe situations.

This uncommon behavior identification is relevant also for testing a CPS. Within the context of this paper, especially the challenges to identifying unpredictable corner cases and coping with environmental complexity [2] are of particular interest. Such unexpected corner cases could be often a cause of failures. However, developers are unable to predict all possible conditions and interaction scenarios in complex environments. Large Language Models (LLMs) and AI assistants show promising support for speeding up development (acceleration mode) or for exploring possible options (exploration mode) [5]. The latter is of particular interest: since LLMs can provide the information they have learned from online sources, one may assume that they know a lot about human behavior. In particular, sources of information about uncommon human behavior are abundant on the Internet (e.g., in newspapers).

Within this paper, we aim to answer the research question of *whether LLMs are effective in proposing human behavior for CPS interaction scenarios.* We propose a systematic approach for exploring human-CPS interaction called SEED (Scenario Elicitation Enhanced

by Large Language moDels). This includes a human behavior ontology describing human-CPS interaction components and different types of human behavior, such as uncommon, unexpected, contradictory, law-breaking, and desired behavior. We use this ontology to tailor prompt templates for LLMs and receive possible human-CPS interaction suggestions. In a preliminary experiment, we have evaluated the usefulness of the suggestions and if they were outside-the-box ideas. The results showed high usefulness and around half of the evaluated outputs as outside-the-box. These suggestions can thus be used to create different detailed scenarios for a use case, taking into account more variants of human behavior.

The paper is structured as follows: section 2 provides relevant background about CPS, modeling human behavior, scenarios and LLMs in requirements engineering. Section 3 describes our vision of how to use LLMs to discover unknown human behavior. Section 4 provides details about the approach and section 5 shows the realization of it using a smart home energy management system as use case. Section 7 discusses our approach and section 8 presents related work. The last section concludes our study.

2 BACKGROUND

We present the background needed to understand our approach: ontologies, scenarios in requirements engineering and LLMs.

Ontologies. An ontology is, informally, an engineering artefact in computer-processable format, which contains the entity types, their relationships, and properties or constraints that hold over them of a particular subject domain (for a longer, more precise definition, see [18]). A practical key difference with conceptual data models is that an ontology is expected to be usable across multiple applications, rather than be tailored to one specific application only. While ontologies were initially envisioned to facilitate data integration, meanwhile they have found use in other computational tasks, such as information retrieval, question-answering systems, Multiple Choice Question generation and user modelling for classifying students, and knowledge discovery [24] and, most recently, they are also used with LLMs (see related work section 8).

Scenarios in Requirements Engineering. Requirements engineering is a key challenge for software engineering [6], as well as CPS engineering [16]. Requirements are not only used to build software systems but also to test them, for instance, as advocated by the popular V-model [35]. User scenarios, which are stories of how users and systems interact to achieve a goal [3], are a common mechanism to capture requirements [15]. They are typically specified by domain experts in the form of natural language text [27], and collected via traditional data collection methods, such as interviews and focus group discussions [28]. This is a slow and costly approach, and therefore there has been considerable effort in optimizing requirements elicitation using crowd-sourced data, such as online resources, and relying on automated techniques, such as natural language processing and machine learning [28]. Requirements in natural language text are often followed up with a substantial manual effort to design and test systems. At the same time, research aims at developing automated techniques for deriving more formal models following a model-driven engineering (MDE) approach (refer to, e.g., [25] for a systematic review).

LLMs. In language processing, traditional language models have been essential for text generation and understanding, employing the simple idea of estimating the likelihood of sequences of tokens based on a training dataset, and predicting the next token given some text (e.g., plain English [45]). Advances in computational power, machine learning, and access to large datasets have led to the rise of LLMs [39]. These models, trained on vast and diverse data, excel at simulating human language. The transformer architecture, introduced by Vaswani et al. in 2017 [48], is the foundation of modern LLMs. The self-attention mechanism and encoder/decoder layers lead to high effectiveness in natural language processing (NLP) tasks that handle long-range dependencies well and parallelize training. By learning from massive corpora and generating realistic text, LLMs are narrowing the gap between human and machine-produced language, not only for NLP but for software engineering [19] and beyond.

3 BIG PICTURE

Traditionally, the development of CPS systems involves a rigorous requirements specification phase, which includes the elicitation of interaction scenarios that describe how the system is expected to react when interacting with its environment, including humans. As it is difficult for developers to come up with a broad variety of situations, our approach supports them by proposing different types of human behavior such as uncommon, unexpected, contradictory, law-breaking, and desired behavior. Following a model-driven engineering philosophy, these requirements are then refined to prescriptive models that specify how the system should be realized in terms of algorithms and implementation technology.

At a high level, our proposed approach is simple (see Figure 1). We suggest using LLMs in conjunction with a human behavior ontology to 1) assist the CPS Developer in creating comprehensive human-CPS interaction scenarios that take into account all forms of unexpected and exceptional human behavior, and 2) help the CPS Developer in transforming the augmented interaction scenarios into prescriptive models and tests for the CPS implementation, as well as descriptive models of the human that the CPS can use at runtime. Our proposed extensions to the current state-of-practice for CPS development are highlighted in blue in Figure 1. The ontology contributes to both structuring types of behavior and typical elements of a CPS and therewith contributes to systematically collecting scenarios comprehensively. The LLM is expected to contribute to the ideation phase and offer creative CPS interaction scenarios that are expected to at least assist humans in devising such scenarios. In this paper, we focus on the LLM and ontology-augmented elicitation of interaction scenarios (bold blue arrows).

Prescriptive Models for CPS. Because CPS are sometimes safety-critical systems, they must follow a rigorous software development process that typically involves certification. MDE plays a critical role in this process: models bridge the gap between requirements and implementation, introducing additional steps and layers of abstraction between the specification and the solution. At each step, the models can be shown to exhibit certain desired properties.

Typically, system interaction scenarios initially take the form of structured text, such as user stories or use cases [21]. In the

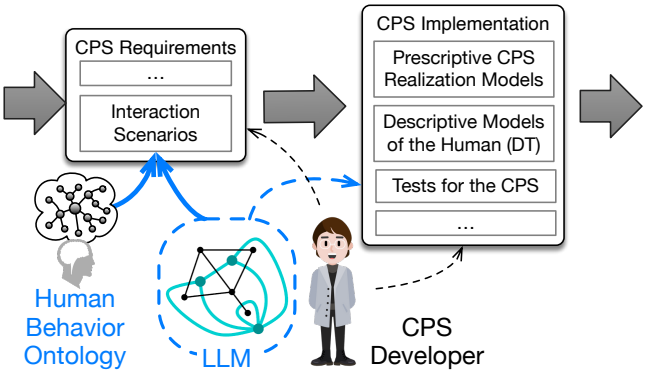

Figure 1: Overview of the LLM-mediated approach, with optional use of an ontology.

context of CPS development, variants of use cases have been proposed that impose additional structure and rigor on how those texts are written. For example, approaches such as *Restricted Use Case Models* (RUCM and U-RUCM [53]), *Concern-Oriented Use Cases* [26] or *UCM4IOT* [8] require the developer to specify the scenarios using flows, where each flow is a numbered sequence of interaction steps. An interaction step is described by one sentence adhering to a simple structure that clearly identifies the interaction kind, direction, and involved actors. This additional rigor makes it possible to then transform those textual scenarios into more formal models, which can then serve as formal specifications that can be analysed and used to derive prescriptive models for the CPS implementation. For example, rigorous use case specifications have been transformed to system operation specifications with formal pre- and post-conditions expressed in OCL [44], probabilistic state machines to assess dependability [56], as well as UML state machines [51].

Descriptive Models of the Human. For describing human behavior and human-system interaction, typical approaches are behavior modeling methods. These models mainly come from requirements engineering processes intending to understand human behavior and human-system interaction. Typical requirements engineering approaches capture such information, e.g., in scenario descriptions within Personas [23], UML activity diagrams, interaction diagrams, BPMN [40] models or domain-specific languages [36]. When it comes to exceptions or exceptional behavior, languages such as BPMN already include concepts to describe this. Other modeling approaches, e.g., rely on adding OCL constraints and checking them [46]. With our envisioned approach, we can add alternative behavior to such models making them more comprehensive as they consider interactions beyond expected behavior. These behavior models could then either be interpreted by a software system, e.g., to provide human behavior assistance [38], or used to further help specify how a system should be tested [30].

Test Generation. One use of human behaviors discovered with LLMs is to support the testing of CPSs. Such testing can be performed in two possible ways. First, the discovered human behaviors can be used as test case specifications for manual testing, i.e., a human reads the test specification and manually performs test actions on the CPS, such as turning on and off switches or tampering with the device. In addition, the discovered human behaviors can be used to support automated testing, which may require translating generated textual specifications into different notations (e.g., restricted natural language models) followed by generating executable test cases. Alternatively, a possibility also exists to translate textual specifications directly into executable test cases, e.g., using LLMs, which requires further investigation. In the case of automated testing, test execution would require testing in simulation.

4 PROPOSED APPROACH

We zoom in on three key components of the first steps of the big picture introduced in the previous section: the human behavior types, the LLM prompting procedure, and prompt templates.

4.1 Human Behavior Types for CPSs

As reviewed in the related work (Section 8), only some notions and terms may be of use for a behavior ontology in the context of human interaction with a CPS. As a first step toward an ontology, we identify broad categories, which may be refined at a later stage, if deemed promising. This list was devised by the authors (with input also from Shaukat Ali), based on their knowledge in the field of human-computer interaction and of CPSs and on consultation of CPS documentation. Behavior in this context refers to a sequence of actions and interactions by an actor with the CPS in a certain context.

- *Expected behavior* concerns behavior (and the corresponding execution of typical actions in concordance with that behavior) that are common and conform to what the CPS is designed for and deemed regular, safe, use of the CPS. For instance, one is expected to cycle with an e-bike.
- *Uncommon and exceptional behavior* are those behaviors that are statistically in the 'long tail', but that the CPS designers have taken into account as valid 'corner cases' and where the CPS will operate within its safety standards.
- *Unexpected behavior* are those behaviors that are statistically in the 'long tail' (and possibly reported on in news articles more often, possibly skewing an LLM trained on it), but which were not foreseen by its designers. The CPS may still operate according to specifications when processing the unexpected behavior.
- *Contradictory behavior* includes a set of behaviors with their related actions where the human indicates one thing but does another. It may or may not be the case that the human should be doing it, nor that the CPS was designed for it. For instance, a user may use the left indicator on a scooter but drive straight or turn right instead of turning left.
- *Desired/intended behavior* is that behavior that the CPS was designed to instill in the user. This is likely also expected and law-abiding, and typically involves 'nudging' of the user towards the desired behavior.
- *Law-abiding behavior* as a type is orthogonal to the aforementioned types of behavior, as it may be either expected or desired.
- *Law-breaking behavior* is likewise orthogonal, but may apply more likely to uncommon, unexpected, or contradictory,

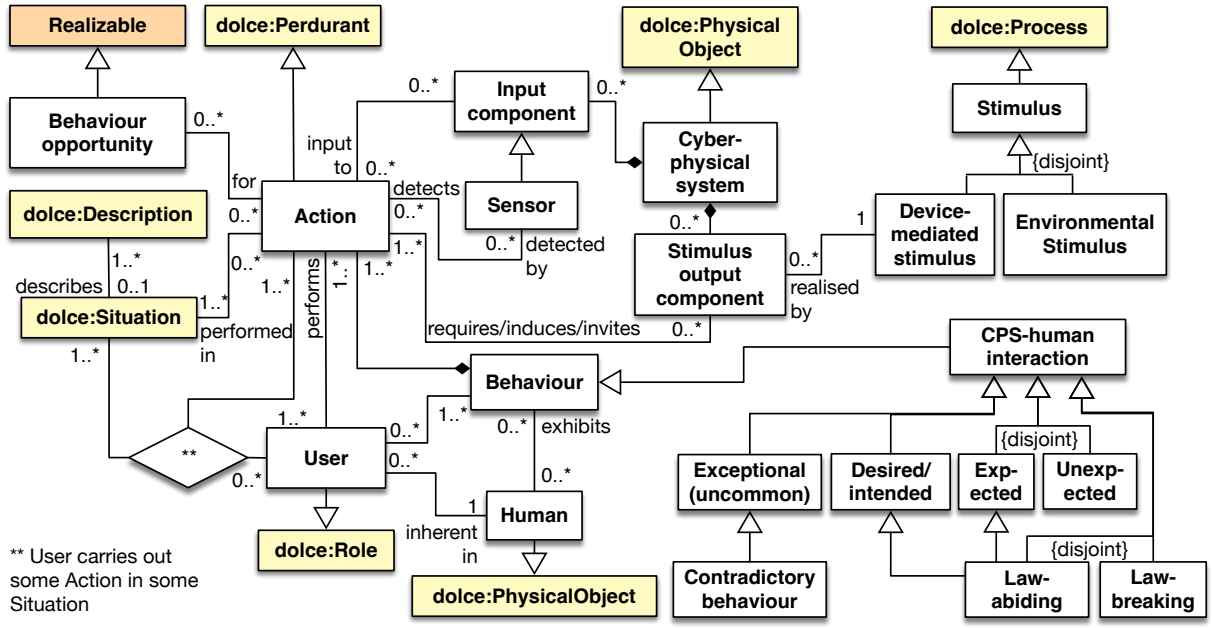

Figure 2: Sketch of the CPS-Human Interaction Behavior Ontology (CHIBO) with the behavior components and attendant entities in the context of CPSs. White boxes: specific to CHIBO; colored boxes: top-level ontology entities for interoperability.

behavior, although the expected category cannot be ruled out, e.g., riding through a red light on an empty road at night.

The notion of *change in human behavior*, can be understood in two ways for CPSs: 'nudging' is effectively taking place or the human's behavior is changing because of non-CPS induced reasons (e.g., from relaxed driving to exhibiting road rage). They are beyond the current scope of the paper.

These considerations resulted in a preliminary (i.e., for experimentation) ontology about human behavior when they interact with CPSs, called CPS-Human Interaction Behavior Ontology (CHIBO), which is sketched informally in Figure 2, rendered diagrammatically for communication and formalized in OWL as the authoritative reference (see fn. 1). CHIBO aims to include the main elements in an overarching structure, where most classes have subclasses tailored to the type of CPS, such as User with as subclasses End user or Technician, and all Actions can be represented in a taxonomy of types of actions and as sequences as being constitutive of a certain type of behavior. Also, while the disjointness between expected versus unexpected behavior is theoretically clear, practically it may be contentious to classify and the constraint may need to be relaxed. Finally, it is aligned to the DOLCE foundational ontology [34], indicated in yellow, and Realizable from BFO [4] that DOLCE does not have. They are included only to improve model quality and precision, interoperability, and potential for reuse.

The ontology can be either extended or a module added for a specific type of CPS, such as the types of sensors and input components, types of users, types of interactions, or one can convert it into a database format where such elements are stored as values. If proven useful, one also may consider adding other ontologies for a specific CPS or generic related models, such as the SASO lightweight application ontology for sensors [22].

4.2 Prompting Procedure

```
cps ← createCPSDescription()*;
humanActors ← populateHumanActors()*;
mainScenarios ← populateMainScenarios(humanActors)*;
configuredOntology ← configureOntology();
for ms : mainScenarios do
    for hb : HumanBehaviorTypes do
        alternateList ← queryLLMForExtraBehavior
            (ms, hb, cps, configuredOntology);
        for alt : alternateList do
            detailedAltScenario = queryLLMForDetails
                (alt, ms, hb, cps);
            ms = integrateAltScenario
                (detailedAltScenario, ms);
        end
    end
end
```

Algorithm 1: Human Behavior Exploration.

The pseudo-code in Algorithm 1 describes the proposed prompting procedure. To start, we need a general description of the CPS, a list of human actors, and the main success interaction scenarios, i.e., a numbered list of steps that explain how the human actors would normally interact with the CPS to achieve their goals. We also need to instantiate the ontology for the CPS under study, e.g.,

specific behavior opportunities. Even for those this initialization, the assistance of an LLM could be used.

Our approach then goes through all the main scenarios, and generates a prompt using the prompt templates (either just based on the human behavior types, or the instantiated ontology). It is this step of the algorithm (i.e., queryLLMForExtraBehavior) that the rest of this paper focusses on. For each newly discovered, relevant behavior, our approach then elicits how the CPS should behave in such a situation by eliciting the details of an alternate scenario.

4.3 Prompt Templates

Consider now the model as shown in Figure 2. For a *simple* prompting strategy (subsequently referred to as SEED-s), we have a template that only contains one placeholder dedicated to capturing the behavior type. This template is: "Given this use case and its main scenario, what is [X] behavior of the user?'. The placeholder [X] is replaced by any of the subtypes of CPS-human interaction behavior.

A more advanced prompting strategy is facilitated by the CHIBO ontology (subsequently named SEED-o). Informed by vocabulary from CHIBO, there are many possible prompts that can be generated by combining the different types. In this paper, we created the following templates:

T1. "Given this use case and its main scenario, what is [X] behavior of user [Y]?" where [X] is replaced with either of the subtypes of CPS-human interaction behavior and [Y] with user or one of its subtypes;
T2. "Given this use case and its main scenario, what is [X] behavior of user [Y] considering behavioral opportunity [Z]?"
T3. "Given this use case and its main scenario, what is [X] behavior of user [Y] when taking into account stimulus [Z]?"
T4. "Given this use case and its main scenario, what is [X] behavior of user [Y] when stimulus output component [Z] malfunctions?"
T5. "Given this use case and its main scenario, what is [X] behavior of user [Y] when action [Z] is obstructed?"

5 RUNNING EXAMPLE: SMART BUILDING

The use case we apply our approach to is taken from the smart building domain. The term smart building refers to both technical processes and systems for the automation and networking of buildings, as well as appropriately equipped buildings. Based on their building topologies, e.g., smart homes, residential buildings, office buildings, data centres and hotels, different scenarios are relevant for human users when interacting with the building and its IT systems. In this paper, we focus on use cases for Smart Homes, informed by documentation of the EU project FINSENY [14].

A Smart Home has a Home Energy Management System. It manages all devices that have a direct or indirect impact on the energy input and output of the smart home. Such devices with impact are (a) all appliances/apparatuses that consume, generate, or store energy, (b) the components of the home, such as walls and windows that regulate the exchange of energy between the inside and the outside, as well as (c) subsets of the home, such as floors or rooms that make sense as separate units for managing energy.

The Smart Home is integrated into an energy distribution network through a 2-way interface that allows information and power to flow in both directions, from the grid to the home (downstream control information & power consumed from the grid by the home) and from the home to the grid (upstream status data & locally stored or generated power fed by the building to the grid). The grid does not have to be aware of the details of the individual appliances and pieces of equipment handled at the home level, as only aggregate information is being exchanged through the interface.

To solve peak loads on the grid, each Energy Provider may contract with a number of its Customers who live in Smart Homes to agree to have some of their appliances interrupted a few hours per year, with the condition that the Customer can derogate at any time. In exchange, the Customer is rewarded financially for participating.

Ontology Instantiation and Exemplary Prompts. Considering the experimental nature of the pipeline, we opted for manual use of the ontology first, rather than automating prompt generation with the OWL file extended with entities relevant to the use case. To this end, we created a spreadsheet with the key classes to be instantiated (or subclasses) as column headings to which all authors could add items taking as the starting point the use case description. This resulted in 27 entities specific to the smart home energy use case, such as Customer and Energy Manager as types of User, Mobile Phone (for SMS) as a type of Stimulus Output Component, and Load Reduction Notification as Device-mediated Stimulus. This spreadsheet can be accessed from our GitHub repository [1].

Let us present here some prompts generated by our SEED-o approach after instantiating the ontology for the second template (Sect. 4.3): "Given this use case and its main scenario, what is unexpected behavior of the user considering they have a financial behavior opportunity?" or, even more specific, assuming a suitable taxonomy for each main class: "Given this use case and its main scenario, what is unexpected behavior of the energy manager considering they have a financial incentive to minimize discounts?". For the fourth template, a specific instance may be "Given this use case and its main scenario, what is law-breaking behavior of the home dweller when the derogation button malfunctions?".

6 EMPIRICAL EVALUATION

In this section we present the empirical evaluation that we have carried out to evaluate our approach. The goal is to answer the following research question:

RQ. Can LLMs effectively assist engineers to identify exceptional human behavior during the requirement elicitation process?

6.1 Experiment Design and Setup

The experiment consisted of an off-site and asynchronous evaluation procedure where the participants had to fill a spreadsheet providing their opinions on the performance of the LLM-generated outputs for the the smart building example.

This spreadsheet contained 82 potential exceptional behaviors generated by the LLM. For each potential exceptional behavior, the participants needed to give their opinion on whether: 1) the suggested exceptional behavior was useful or not, and 2) whether they thought it was outside-the-box or not (meaning something they

would not have thought of themselves). They also were requested to identify repeated suggested behaviors.

The way in which the spreadsheet was created is as follows. We created prompts for our two prompting strategies: the simple strategy and one that uses the CHIBO ontology, as described in Section 4.3. For the simple strategy, we created 4 prompts (prompts type a). For the SEED-o strategy, we have created prompts using the instantiation of the ontology (Section 5) which led to 21 prompts (prompts type b). We have prompted GPT-4o with all these 25 prompts and, for each prompt of type a, we made the LLM generate between 10-12 potential exceptional behaviors; and 4-5 potential exceptional behaviors for the prompts of type b. In all cases, we provided a system prompt that contained a description of the smart building and a main success scenario. The ontology, all the prompts, and responses we have used in this experiment are available in our GitHub repository[1].

For instance, for the prompt "Given this use case and its main scenario, what is law-breaking behavior of the customer considering she wants to maximize electricity use?", GPT-4o provides, among others, the following potential exceptional behavior "Tampering with the Home Energy Management System: Illegally altering or hacking the Home Energy Management System to disable load reduction commands or to falsify energy usage data. This could involve: - Hacking into the system to prevent it from turning off high-energy appliances. - Reprogramming the system to falsely report lower energy usage during the load reduction period.".

To have a balanced list of exceptional behaviors between the two strategies, we took 10 behaviors for each prompt type a (40 suggestions) and 2 behaviors for each prompt type b (42 suggestions). We put them together in the spreadsheet and shuffled them.

The preparation of the experiment was done by one author. The other four authors, having no knowledge of how the spreadsheet was generated (e.g., the strategy that generated every potential exceptional behavior) beyond the example and the main scenario, participated by each filling the spreadsheet independently.

Filling the spreadsheets took each author around 2 hours, mainly due to the time-consuming detection of duplicate behaviors. Once filled in, the four spreadsheets were handed in to the author who prepared the experiment in order to analyze the results.

6.2 Evaluation Metrics

To analyze the effectiveness of the LLM, we defined the following metrics:

- **M1** Usefulness: Proportion of useful behaviors identified by each strategy,
- **M2** Outside-the-box: Proportion of outside-the-box behaviors identified by each strategy,
- **M3** Duplicates: The proportion of duplicated behaviors per strategy,
- **M4** Overlap: The overlap between the behaviors suggested by the two strategies.

6.3 Results

Figure 3 shows the results of M1 and M2 for each participant and each strategy. As we can observe, all the participant seem to agree

[1]https://github.com/atenearesearchgroup/human-behavior-exploration

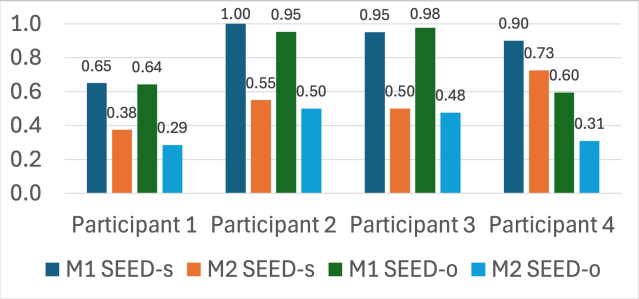

Figure 3: Results for M1 (usefulness) and M2 (outside-the-box) of the evaluated output, calculated as a fraction of 1 from the Y(1)/N(0) answers.

that the suggestions are useful (M1). The average opinion of all four participants on the usefulness of the simple approach is 0.88 with a standard deviation of 0.13, while the average usefulness of the CHIBO approach is 0.79 ± 0.17. The participants also considered that there is a good number of suggestions that are outside-the-box. The average outside-the-box suggestions for the simple approach is 0.54 ± 0.13, while for the CHIBO strategy, it is 0.39 ± 0.10.

Figure 4 presents the results for M3 and M4. We can observe that the participants considered that our approach provides some duplicated behaviors for SEED-s (on average 0.18 ± 0.09) and even a higher number for CHIBO (0.38 ± 0.09). Regarding the overlap between the exceptional behaviors suggested by both strategies, we can observe there is a certain overlap. According to the participants, 0.36 ± 0.16 elements proposed by the simple approach are also suggested by the SEED-s strategy, and 0.49 ± 0.18 of the behaviors proposed by the SEED-o strategy are suggested by the simple strategy, too. However, these numbers are far from 1, hence we can conclude that both strategies seem to complement each other.

An example of complementarity is SEED-s's output "**Multiple Users in the Home**: Other members of the household might not be aware of the load reduction agreement and could operate appliances in ways that contradict the intended load reduction, resulting in unexpected energy usage." that did not appear in SEED-o's output and, vice versa, "**Communication Failures**: This can result in the Provider failing to update the customer's records, which might affect the calculations for financial rewards or penalties." in SEED-o but that did not appear in SEED-s.

7 DISCUSSION

The results showed that both the basic and ontology-mediated prompting resulted in useful and outside-the-box outputs, in essence exploiting a strength of LLMs, being that of creativity and ingested corner cases reported on the Web. Based on a preliminary exploration of ontology-mediated prompting resulting in more creative suggestions than the basic one, the expectation was that it would also hold in the experiment, but it did not. This noted absence of difference may be due to having selected only the first two responses from the list of ontology-mediated responses for the evaluation; see supplementary material for complete outputs (see fn. 1). For instance, "Given this use case and its main scenario, what

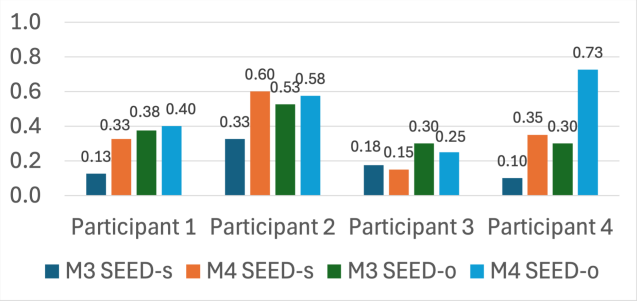

Figure 4: Results for M3 (duplicates) and M4 (overlap) in the outputs evaluated.

is unexpected behavior of the customer when the derogation is obstructed?" takes into account communication failures and users that the SEED-s does not, such as "The Home Energy Management System might successfully receive the opt-out or cancellation request, but it fails to inform the Distributor and subsequently the Provider.". This scenario was listed third and therefore not included in the user evaluation, and likewise other themes further down in the responses, such as fake documents, technical assistance, and the use of HVAC equipment did not appear in the SEED-s output.

The differences across the four participants may be due to the participants' respective background, both regarding familiarity with smart home electricity meters and society. For instance, tampering, bypassing, and illegal tapping are common in the country where one of the participants lives, as are the many notifications about involuntary load-shedding (rolling blackouts), and so the assessment of useful or outside-the-box may be affected accordingly. Likewise, assumptions and familiarity with a home energy system can affect the judgments, as well as the terminology (e.g., whether the 'home energy system' refers to the software only or also all hardware components including the meter and cabling). This brings afore a potential new avenue of future work for scenario specification, requirements engineering, and test specification: the notion of multiple stakeholders judging the relevance of the draft scenarios before finalizing the requirements and test specifications.

Regarding the SEED configuration and especially the positioning of the ontology and the LLM, other options are conceivable, most notably embedding the ontology in the LLM (pre-training stage of the LLM) or including it in an enhanced prompting stage with already known outputs (fine-tuning of the LLM) that are not readily available in our case. Both options require considerable upfront investment from stakeholders in both resources and knowledge and skills to carry out, which are prohibitive for uptake of the approach. The stakeholders' strengths are in CPS and requirements engineering, rather, but even if that is addressed, it does not resolve the issue of limited scenarios to train an LLM on. In contrast, our method offers a flexible approach with comparatively low upfront investment that leverages an existing strength of LLMs and, as has been shown, already offers useful output.

Finally, although the running example and focus in this paper is CPS, the general approach of LLM-mediated 'uncommon' case exploration may be generalisable to requirements elicitation in general, not just for a CPS, and be used to complete interaction scenarios that will result in more robust implementations.

8 RELATED WORK

We focus related work in the pertinent aspects of LLMs, both retarding ontology-enhance LLM use and their use in requirements engineering. Afterward, we will briefly discuss the rationale why a new model about CPS-Human Interaction behavior had to be developed to tailor the prompts.

Ontology-enhanced LLM use. The use of ontologies with LLMs is not novel, especially in the context of knowledge graph and ontology learning and RAG. Some titles also suggest use of ontologies to enhance prompting, such as [29, 41, 50, 55], but they use it in different tasks than aimed at here (model-driven prompting as a systematic approach to explore a theme), such as few-shot learning to patch up perceived knowledge gaps [50], towards information retrieval for a dialog system [41], for stance detection task [55], and as background knowledge for event detection through causal reasoning [29].

LLMs in Requirements Engineering. LLMs have enjoyed increasing popularity in software engineering research, covering many different areas of software engineering ranging from requirements to testing and maintenance [13, 19]. LLMs for requirements engineering, while amounting to a relatively smaller portion of existing work — 3.9% of the surveyed studies by Hou et al. [19] and also remarked by Fan et al. [13], address topics including anaphoric ambiguity treatment, requirements classification, requirements analysis and evaluation, and specification generation [19]. Related work typically takes existing requirements as a starting point and performs various analyses or transformations on top of them using LLMs. Examples include requirement retrieval on requirement analysis tasks [52], automatic requirement classification [31], requirement simulation and disambiguation [49] and specification generation/-formalization from input Java code [32] or natural language text intent [12], and generating use cases [54] or domain models [9] from natural language descriptions.

Besides these, to the best of the authors' knowledge, there is no other comparable work to ours using LLMs for requirements elicitation from scratch, as a means of capturing human behavior from the collective knowledge of the LLMs. We believe this to be an important gap in the literature and fulfilling a high-level design goal in CPS research but also in general. This is in line with Fan et al., who report a lack of volume and even a reluctance to use LLMs for higher-level design goals [13].

Models about human behavior. Before developing the basic ontology of CPS-Human Interaction Behavior, we examined related literature and, where available, the OWL files of the ontologies. Blanch et al. [7], reviewed 17 ontologies of human behavior in a broad sense, which covered principally neuroscience, phenotypes, psychiatry and related health, which vary widely in size (ranging from 110 to over 100K classes), rather than the behavior of humans when they interact with CPSs specifically. Two of them do contain relevant content, being the Cognitive Paradigm ontology (CogPo) and Emotions & Cognition ontology (ECO), and most recently the

behavior Change Intervention ontology (BCIO) was released. We will highlight the pertinent content of each.

The CogPo [47] is aligned to the BFO top-level ontology and a shallow list of terms with few relations. Key entities are types of stimuli (a.o., Braille dots, heat, vibratory stimulation, etc.) and stimulus modality (e.g., auditory, tactile), and a long list of behavioral Experimental Paradigms mostly unrelated to CPSs. The ECO [17] has content on sensors, contexts, and motivation, and relations between them and on paper it is aligned to DOLCE with its D&S extension. Regarding behavior, it contains only Neutral behavior and NonNeutral behavior, but no typology for what constitutes 'non-neutral'. The BCIO [33] is also aligned to BFO and contains many distinct high-level topics rather than concrete content relevant to CPS interactions and our purpose is not behavior manipulation. Generally usable themes include goals and actions, and its financial behavior opportunity may be recast as relevant for (almost) law-breaking behavior, such as stealing, and Temporal behavior opportunity could be re-cast as a user taking too much time or be in a hurry in its interaction with a CPS.

Other models that might be of use are the fuzzy ontology for "human activity representation" to recognize human behavior [11] and human–CPS integration patterns [43]. The former is unrelated to CPSs and we do not need the fuzzy reasoning. The human–CPS interaction patterns in Tables 2 and 3 of the study [43] includes typical common roles that are of general use, such as engineer and technician.

In sum, no extant model or ontology is about human behavior as they interact with CPSs, and thus also not regarding requirements generation and scenario creation for CPSs, albeit that some of the stimuli of CogPo may apply as well as the few types of behavior found, as prospective terms.

9 CONCLUSION

We proposed a preliminary method, called SEED, that incorporates an LLM to elicitate input for scenarios tailored to cyber-physical systems. This has been shown to provide additional information about uncommon, yet realistic, human-CPS interaction behavior that may not be easy to discover by typical human stakeholders. The initial user evaluation for a home energy management system use case showed that the simple and ontology-mediated prompt variants of our approach complement each other to increase diversity in interaction scenarios. It is expected that this eventually will result in a more robust CPS implementation.

Future work includes a more extensive evaluation, assessment of a larger output set, and prospective automation of SEED.

Acknowledgments

We are grateful to Shaukat Ali for his input in the preliminary scoping and discussion of the work.
This work has been partially supported by Junta de Andalucía under project QUAL21 010UMA, by the Spanish Government (FEDER, Ministerio de Ciencia e Innovación–Agencia Estatal de Investigación) under projects TED2021-130523B-I00 and PID2021-125527NB-I00, and Universidad de Málaga under project B1-2022_81. This work has been partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2023 Internet of Production - 390621612. Website: https://www.iop.rwth-aachen.de/.

REFERENCES

[1] 2024. Github Repository. https://github.com/atenearesearchgroup/human-behavior-exploration.
[2] Afsoon Afzal, Claire Le Goues, Michael Hilton, and Christopher Steven Timperley. 2020. A Study on Challenges of Testing Robotic Systems. In *2020 IEEE 13th Int. Conf. on Software Testing, Validation and Verification (ICST)*. 96–107.
[3] Ian F Alexander and Neil Maiden. 2005. *Scenarios, stories, use cases: through the systems development life-cycle.* John Wiley & Sons.
[4] Robert Arp, Barry Smith, and Andrew D. Spear. 2015. *Building Ontologies with Basic Formal Ontology.* The MIT Press, USA.
[5] Shraddha Barke, Michael B. James, and Nadia Polikarpova. 2023. Grounded Copilot: How Programmers Interact with Code-Generating Models. *Proc. ACM Program. Lang.* 7, OOPSLA1 (2023). https://doi.org/10.1145/3586030
[6] Christoph Becker, Stefanie Betz, Ruzanna Chitchyan, Leticia Duboc, Steve M Easterbrook, Birgit Penzenstadler, Norbet Seyff, and Colin C Venters. 2015. Requirements: The key to sustainability. *IEEE Software* 33, 1 (2015), 56–65.
[7] Angel Blanch, Roberto García, Jordi Planes, Rosa Gil, Ferran Balada, Eduardo Blanco, and Anton Aluja. 2017. Ontologies About Human Behavior. *European Psychologist* 22, 3 (2017), 180–197. https://doi.org/10.1027/1016-9040/a000295
[8] Paul Boutot, Mirza Rehenuma Tabassum, Abdul Abedin, and Sadaf Mustafiz. 2024. Requirements development for IoT systems with UCM4IoT. *Journal of Computer Languages* 78 (2024), 101251. https://doi.org/10.1016/j.cola.2023.101251
[9] Fatma Bozyigit, Tolgahan Bardakci, Alireza Khalilipour, Moharram Challenger, Guus Ramackers, Önder Babur, and Michel RV Chaudron. 2024. Generating domain models from natural language text using NLP: a benchmark dataset and experimental comparison of tools. *Software and Systems Modeling* (2024), 1–19.
[10] Marius Brinkmann and Axel Hahn. 2017. Testbed architecture for maritime cyber physical systems. In *2017 IEEE 15th Int. Conf. on Industrial Informatics (INDIN)*. 923–928. https://doi.org/10.1109/INDIN.2017.8104895
[11] Natalia Díaz Rodríguez, Manuel P. Cuéllar, Johan Lilius, and Miguel Delgado Calvo-Flores. 2014. A fuzzy ontology for semantic modelling and recognition of human behaviour. *Knowledge-Based Systems* 66 (2014), 46–60. https://doi.org/10.1016/j.knosys.2014.04.016
[12] Madeline Endres, Sarah Fakhoury, Saikat Chakraborty, and Shuvendu K Lahiri. 2023. Formalizing Natural Language Intent into Program Specifications via Large Language Models. *arXiv preprint arXiv:2310.01831* (2023).
[13] Angela Fan, Beliz Gokkaya, Mark Harman, Mitya Lyubarskiy, Shubho Sengupta, Shin Yoo, and Jie M Zhang. 2023. Large language models for software engineering: Survey and open problems. *arXiv preprint arXiv:2310.03533* (2023).
[14] FINSENY 2024. FINSENY (Future INternet for Smart ENergY). http://www.finesce.eu/FINSENY.html
[15] Samuel A Fricker, Rainer Grau, and Adrian Zwingli. 2014. Requirements engineering: best practice. In *Requirements Engineering for Digital Health*. Springer, 25–46.
[16] Eva Geisberger and Manfred Broy. 2012. *agendaCPS: integrierte forschungsagenda cyber-physical systems*. Vol. 1. Springer-Verlag.
[17] Rosa Gil, Jordi Virgili-Goma, Roberto García, and Cindy Mason. 2015. Emotions ontology for collaborative modelling and learning of emotional responses. *Computers in Human Behavior* 51 (2015), 610–617. https://doi.org/10.1016/j.chb.2014.11.100 Computing for Human Learning, Behaviour and Collaboration in the Social and Mobile Networks Era.
[18] Nicola Guarino, Daniel Oberle, and Steffen Staab. 2009. What Is An Ontology? In *Handbook on Ontologies*. Springer, Chapter 1, 1–17.
[19] Xinyi Hou, Yanjie Zhao, Yue Liu, Zhou Yang, Kailong Wang, Li Li, Xiapu Luo, David Lo, John Grundy, and Haoyu Wang. 2024. Large Language Models for Software Engineering: A Systematic Literature Review. arXiv:2308.10620 [cs.SE]
[20] Jerome Hugues, Anton Hristosov, John J. Hudak, and Joe Yankel. 2020. TwinOps - DevOps meets model-based engineering and digital twins for the engineering of CPS. In *23rd ACM/IEEE Int. Conf. on Model Driven Engineering Languages and Systems: Comp. (MODELS '20)*. ACM. https://doi.org/10.1145/3417990.3421446
[21] Ivar Jacobson, Magnus Christerson, Patrik Jonsson, and Gunnar Overgaard. 1992. *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison Wesley.
[22] Krzysztof Janowicz, Armin Haller, Simon J.D. Cox, Danh Le Phuoc, and Maxime LefranÃ§ois. 2019. SOSA: A lightweight ontology for sensors, observations, samples, and actuators. *Journal of Web Semantics* 56 (2019), 1–10. https://doi.org/10.1016/j.websem.2018.06.003
[23] Devi Karolita, Jennifer McIntosh, Tanjila Kanij, John Grundy, and Humphrey O. Obie. 2023. Use of personas in Requirements Engineering: A systematic mapping study. *Information and Software Technology* 162 (2023), 107264. https://doi.org/10.1016/j.infsof.2023.107264
[24] C. Maria Keet. 2018. *An introduction to ontology engineering*. Computing, Vol. 20. College Publications, UK. 334p.

[25] Shekoufeh Kolahdouz-Rahimi, Kevin Lano, and Chenghua Lin. 2023. Requirement Formalisation using Natural Language Processing and Machine Learning: A Systematic Review. *arXiv preprint arXiv:2303.13365* (2023).
[26] Ryan Languay, Nika Prairie, and Jörg Kienzle. 2023. Concern-Oriented Use Cases. *Journal of Object Technology* 22, 2 (July 2023), 2:1–14. https://doi.org/10.5381/jot.2023.22.2.a13
[27] Christophe Lemaigre, Josefina Guerrero García, and Jean Vanderdonckt. 2008. Interface model elicitation from textual scenarios. In *Human-Computer Interaction Symposium: IFIP 20th World Computer Congress, 1st TC 13 Human-Computer Interaction Symposium (HCIS 2008)*. Springer, 53–66.
[28] Sachiko Lim, Aron Henriksson, and Jelena Zdravkovic. 2021. Data-driven requirements elicitation: A systematic literature review. *SN Computer Science* 2, 1 (2021), 16.
[29] Jintao Liu, Zequn Zhang, Zhi Guo, Li Jin, Xiaoyu Li, Kaiwen Wei, and Xian Sun. 2023. KEPT: Knowledge Enhanced Prompt Tuning for event causality identification. *Knowledge-Based Systems* 259 (2023), 110064. https://doi.org/10.1016/j.knosys.2022.110064
[30] Daniel Lübke and Tammo van Lessen. 2017. BPMN-Based Model-Driven Testing of Service-Based Processes. In *Enterprise, Business-Process and Information Systems Modeling*. Springer International Publishing, Cham, 119–133.
[31] Xianchang Luo, Yinxing Xue, Zhenchang Xing, and Jiamou Sun. 2022. Prcbert: Prompt learning for requirement classification using bert-based pretrained language models. In *37th Int. Conf. on Automated Software Engineering*. 1–13.
[32] Lezhi Ma, Shangqing Liu, Yi Li, Xiaofei Xie, and Lei Bu. 2024. SpecGen: Automated Generation of Formal Program Specifications via Large Language Models. *arXiv preprint arXiv:2401.08807* (2024).
[33] M. M. Marques, A. J. Wright, E. Corker, M. Johnston, R. West, J. Hastings, L. Zhang, and S. Michie. 2023. The Behaviour Change Technique Ontology: Transforming the Behaviour Change Technique Taxonomy v1 [version 1; peer review: 4 approved]. *Wellcome Open Research* 8 (2023), 308. https://doi.org/10.12688/wellcomeopenres.19363.1
[34] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, and A. Oltramari. 2003. Ontology Library. WonderWeb Deliverable D18 (ver. 1.0, 31-12-2003).. http://wonderweb.semanticweb.org.
[35] Sonali Mathur and Shaily Malik. 2010. Advancements in the V-Model. *International Journal of Computer Applications* 1, 12 (2010), 29–34.
[36] Judith Michael and Heinrich C. Mayr. 2013. Conceptual Modeling for Ambient Assistance. In *Conceptual Modeling - ER 2013 (LNCS, Vol. 8217)*. Springer, 403–413.
[37] Judith Michael, Maike Schwammberger, and Andreas Wortmann. 2024. Explaining Cyberphysical System Behavior with Digital Twins. *IEEE Software* 41, 1 (Jan 2024), 55–63. https://doi.org/10.1109/MS.2023.3319580
[38] Judith Michael and Volodymyr Shekhovtsov. 2024. A Model-Based Reference Architecture for Complex Assistive Systems and its Application. *Journal Software and Systems Modeling (SoSyM)* (2024). https://doi.org/10.1007/s10270-024-01157-1
[39] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *arXiv preprint arXiv:2402.06196* (2024).
[40] OMG. 2013. *Business Process Model and Notation (BPMN), Version 2.0.2.* Technical Report. Object Management Group.
[41] Oleksandr Palagin, Vladislav Kaverinskiy, Anna Litvin, and Kyrylo Malakhov. 2023. OntoChatGPT Information System: Ontology-Driven Structured Prompts for ChatGPT Meta-Learning. *International Journal of Computing* (July 2023), 170–183. https://doi.org/10.47839/ijc.22.2.3086
[42] Omid Rajabi Shishvan, Daphney-Stavroula Zois, and Tolga Soyata. 2018. Machine Intelligence in Healthcare and Medical Cyber Physical Systems: A Survey. *IEEE Access* 6 (2018), 46419–46494. https://doi.org/10.1109/ACCESS.2018.2866049
[43] Doruk Sahinel, Cem Akpolat, Orhan Can Görür, Fikret Sivrikaya, and Sahin Albayrak. 2021. Human modeling and interaction in cyber-physical systems: A reference framework. *Journal of Manufacturing Systems* 59 (1 April 2021), 367–385. https://doi.org/10.1016/j.jmsy.2021.03.002
[44] Shane Sendall and Alfred Strohmeier. 2000. From Use Cases to System Operation Specifications. In *UML 2000 - The Unified Modeling Language*. Springer.
[45] Claude E Shannon. 1951. Prediction and entropy of printed English. *Bell system technical journal* 30, 1 (1951), 50–64.
[46] Neelam Soundarajan and Stephen Fridella. 1999. Modeling exceptional behavior. In *2nd International Conference on The Unified Modeling Language: Beyond the Standard* (Fort Collins, CO, USA) *(UML'99)*. Springer-Verlag, 691–704.
[47] J. A. Turner, G. Frishkoff, and Laird A. R. 2011. Ontology harmonization between fMRI and ERP: CogPO and NEMO. In *41th Annual Meeting of the Society for Neuroscience*. Washington D.C., USA.
[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
[49] Jules White, Sam Hays, Quchen Fu, Jesse Spencer-Smith, and Douglas C Schmidt. 2024. Chatgpt prompt patterns for improving code quality, refactoring, requirements elicitation, and software design. In *Generative AI for Effective Software Development*. Springer, 71–108.
[50] Hongbin Ye, Ningyu Zhang, Shumin Deng, Xiang Chen, Hui Chen, Feiyu Xiong, Xi Chen, and Huajun Chen. 2022. Ontology-enhanced Prompt-tuning for Few-shot Learning. In *ACM Web Conference 2022 (WWW '22)*. ACM, 778–787. https://doi.org/10.1145/3485447.3511921
[51] Tao Yue, Shaukat Ali, and Lionel Briand. 2011. Automated Transition from Use Cases to UML State Machines to Support State-Based Testing. In *Modelling Foundations and Applications*. Springer, 115–131.
[52] Jianzhang Zhang, Yiyang Chen, Nan Niu, and Chuang Liu. 2023. A preliminary evaluation of chatgpt in requirements information retrieval. *arXiv preprint arXiv:2304.12562* (2023).
[53] Man Zhang, Tao Yue, Shaukat Ali, Bran Selic, Oscar Okariz, Roland Norgre, and Karmele Intxausti. 2018. Specifying uncertainty in use case models. *JSS* 144 (2018), 573–603. https://doi.org/10.1016/j.jss.2018.06.075
[54] Simiao Zhang, Jiaping Wang, Guoliang Dong, Jun Sun, Yueling Zhang, and Geguang Pu. 2024. Experimenting a New Programming Practice with LLMs. *arXiv preprint arXiv:2401.01062* (2024).
[55] Kai Zheng, Qingfeng Sun, Yaming Yang, and Fei Xu. 2022. Knowledge Stimulated Contrastive Prompting for Low-Resource Stance Detection. In *Findings of the Association for Computational Linguistics: EMNLP 2022*. Assoc. for Computational Linguistics, 1168–1178. https://doi.org/10.18653/v1/2022.findings-emnlp.83
[56] Miriam Zia, Sadaf Mustafiz, Hans Vangheluwe, and Jörg Kienzle. 2007. A Modelling and Simulation Based Process for Dependable Systems Design. *Software and Systems Modeling* (April 2007), 437 – 451.