


## Bachelor's Thesis

<b>Title of Bachelor's Thesis (English)</b>	LLMs for User Assistance in Gathering Domain Knowledge
<b>Title of Bachelor's Thesis (German)</b>	LLMs zur Unterstützung von Nutzer:innen bei der Erhebung von fachspezifischem Wissen
<b>Author (last name, first name):</b>	Hemedinger Jonas Christian
<b>Student ID number:</b>	11933393
<b>Degree program:</b>	Bachelor of Business and Economics, BSc (WU) 
<b>Examiner (degree, first name, last name):</b>	Dipl.-Ing. Katrin Schreiberhuber, B.Sc., Univ.Prof. Marta Sabou, Ph.D.

I hereby declare that:

1. I have written this Bachelor's thesis myself, independently and without the aid of unfair or unauthorized resources. Whenever content has been taken directly or indirectly from other sources, this has been indicated and the source referenced.
2. This Bachelor's Thesis has not been previously presented as an examination paper in this or any other form in Austria or abroad.
3. This Bachelor's Thesis is identical with the thesis assessed by the examiner.
4. (Only applicable if the thesis was written by more than one author): this Bachelor's thesis was written together with

The individual contributions of each writer as well as the co-written passages have been indicated.

05/05/2025

Date

  
signature

Bachelor Thesis

# LLMs for User Assistance in Gathering Domain Knowledge

Jonas C. Hemedinger

Date of Birth: 13.03.1999

Student ID: 11933393

**Subject Area:** Information Business

**Studienkennzahl:** UJ 033 561

**Supervisors:**

Univ.Prof. Marta Sabou, Ph.D.

Dipl.-Ing. Katrin Schreiberhuber, B.Sc.

**Date of Submission:** 05. May 2025

*Department of Information Systems & Operations Management, Vienna University of Economics and Business, Welthandelsplatz 1, 1020 Vienna, Austria*

# Contents

<b>1</b>	<b>Introduction</b>	<b>8</b>
<b>2</b>	<b>Related Work</b>	<b>9</b>
2.1	Cyber-Physical Systems . . . . .	10
2.2	Semantic Technologies . . . . .	10
2.3	Knowledge Elicitation . . . . .	11
2.4	LLMs in Knowledge Elicitation . . . . .	12
2.5	Prompt Engineering . . . . .	12
2.6	User-Centered Design . . . . .	14
<b>3</b>	<b>Methodology</b>	<b>14</b>
3.1	Design and Evaluation Framework . . . . .	15
3.2	Prompt Development Process . . . . .	16
3.3	Prompt Design and Engineering Patterns . . . . .	17
3.3.1	Meta Language Creation Pattern . . . . .	18
3.3.2	Flipped Interaction Pattern . . . . .	19
3.3.3	Persona Pattern . . . . .	21
3.3.4	Cognitive Verifier Pattern . . . . .	21
3.3.5	Fact Check List Pattern . . . . .	22
3.3.6	Reflection Pattern . . . . .	23
3.3.7	Template Pattern . . . . .	24
3.4	Technical Implementation . . . . .	24
3.5	Use Case . . . . .	26
<b>4</b>	<b>Evaluation and Results</b>	<b>27</b>
4.1	Evaluation Criteria and Scoring System . . . . .	28
4.2	Test Scenario . . . . .	30
4.3	Model Evaluation . . . . .	31
4.3.1	Deepseek-r1 (70B) . . . . .	31
4.3.2	LLaMA 3.3 (70B) . . . . .	33
4.3.3	GPT-4 . . . . .	35
4.3.4	GPT-4 Turbo . . . . .	37
4.4	Comparative Analysis of Model Scores . . . . .	38
<b>5</b>	<b>Discussion and Conclusion</b>	<b>40</b>
5.1	Future Work . . . . .	42
5.1.1	Expanded Evaluation Approaches . . . . .	43
5.1.2	Refinement of Scoring System . . . . .	43
5.1.3	Impact of Model Temperature on Prompt Results . . . . .	44



## List of Tables

1	Overview of the seven Prompt Engineering Patterns utilized . . .	18
2	Scoring Rubric for Completeness . . . . .	28
3	Scoring Rubric for Relevance . . . . .	29
4	Scoring Rubric for Adherence to Instructions and Patterns . .	30
5	Comparative Scores of LLMs Across Evaluation Criteria . . .	39

## List of Figures

- 1 Example output of the LLM in CSV-style table format. . . . 24

# Abstract

This thesis explores how prompt-based interaction with large language models (LLMs) can support stakeholders in applying the SENSE guideline for integrating a semantic explainability stack into cyber-physical systems (CPS). The guideline is difficult in its application because of the technical complexity and use of domain specific terminology. To address this, a specifically developed prompt was designed that uses prompt engineering techniques to guide LLMs through a structured support process. The prompt facilitates the step-by-step elicitation of domain knowledge by asking targeted questions, enforcing structured output and applying behavioral rules such as fact-checking, reflection and standardized CSV-style formatting.

The approach was evaluated using four LLMs - Deepseek-r1 (70B), LLaMA 3.3 (70B), GPT-4 and GPT-4 Turbo - along predefined criteria: *completeness*, *relevance* and *adherence to instructions*. Results show that while all models varied in performance, GPT-4 Turbo achieved the highest completeness score, while GPT-4 showed the most balanced overall performance across all evaluation criteria. The findings demonstrate the potential of prompt engineering to enhance LLM-based user assistance for domain knowledge elicitation. This work contributes to bridging the gap between the technical complexity of semantic explainability frameworks and end-user adoption in CPS contexts.

# 1 Introduction

Cyber-physical systems (CPS) enable automation by combining computational elements with physical systems [4]. With the increasing complexity of CPS and the challenges they pose in terms of transparency, explainability has become a critical factor [10]. In order to make well-founded decisions and successfully integrate systems, it is important that stakeholders can understand and trust the behavior of such systems. The SENSE project <sup>1</sup> addresses these challenges by proposing a semantics-based explanation framework for CPS. The project developed an explainability stack that integrates semantic technologies, digital twins, and user-centered interfaces to create explainable cyber-physical systems (ExpCPS) [2] that offer structured, comprehensible, and actionable insights into system behavior. ExpCPS combine sensor-based monitoring, the control of physical processes and the mapping of causal knowledge to make decision-making processes more transparent. This allows stakeholders to understand how individual elements in the system influence the overall result [2].

A central element of the SENSE framework is the use of a knowledge graph (KG), which serves as a structured knowledge base and maps system knowledge and interrelationships. However, the creation and population of such a graph is associated with considerable challenges [12], especially in the area of knowledge elicitation, the process of capturing, structuring and integrating domain expertise [6]. When using the SENSE guideline to integrate the SENSE explainability stack into an existing system, users must enter various attributes in an Excel template including platforms, sensors, their connections, observable properties and specific instances. However, applying the rules of the SENSE guideline to correctly structure and present system information can be complex and difficult to understand.

This work addresses the existing challenges through a structured, prompt engineering-based approach to support the implementation of the SENSE guideline. The focus is on a carefully designed prompt that uses established prompt engineering techniques to guide LLMs through a step-by-step support process. Clear instructions, targeted questions and predefined output structures reduce the complexity of the guideline and enable consistent, structured documentation of results. Outputs are in CSV-style spreadsheet format so that data can be transferred directly into the SENSE Excel template. With a strong focus on usability and comprehensibility, the prompt makes it easy to navigate through the technical complexity of the SENSE framework - even for users with no prior knowledge of semantic technologies.

---

<sup>1</sup>SENSE Project: <https://sense-project.net> (accessed April 15, 2025)



To guide this investigation, the following research questions are addressed:

- *How can existing prompt engineering techniques be synergistically integrated to consistently generate accurate and structured outputs that conform to the SENSE guideline for knowledge graph construction?* [RQ1]
- *To what extent do Deepseek-r1 (70B), LLaMA 3.3 (70B), GPT-4 and GPT-4 Turbo differ in terms of output completeness, relevance, and adherence to instructions and patterns when assisting with knowledge graph construction?* [RQ2]

To answer these research questions, this work relies on the combination of various prompt engineering patterns, which are integrated into a structured prompt. This prompt supports the systematic collection and validation of domain-specific knowledge and outputs the collected data in a standardized CSV format. Through a detailed evaluation of the interaction with the Deepseek-r1, LLaMA 3.3, GPT-4 and GPT-4 Turbo models, the quality of the results obtained is analyzed and assessed in terms of completeness, relevance and adherence to instructions and patterns.

The structure of this thesis is as follows: Section 2 provides an overview of related work, covering key concepts such as CPS, semantic technologies, knowledge elicitation and relevant foundations for LLM use in elicitation processes, including prompt engineering. Section 3 outlines the research methodology, detailing the prompt development process, technical implementation and the set of applied prompt engineering patterns. Section 4 presents the evaluation and results, including the definition of the test scenario, model performance assessments, and a comparison across four LLMs. Section 5 concludes the thesis with a discussion of findings, implications for future work and considerations for enhancing the evaluation framework.

## 2 Related Work

This section summarizes relevant literature and existing research findings in order to provide an understanding of the context of this work and define the theoretical framework. First, fundamental concepts in the field of cyber-physical systems (CPS) are presented. Semantic technologies and their significance for knowledge modeling are then discussed. The focus is also on the challenges and methods of knowledge elicitation, in particular the acquisition and structuring of implicit and domain-specific knowledge. In addition,

relevant prompt engineering techniques that are used to interact with LLMs are presented. Finally, user-centered design principles are discussed in order to emphasize the relevance of a user-centered approach in the design of interactive systems.

## 2.1 Cyber-Physical Systems

Cyber-physical systems (CPS) integrate computational, physical, and communication components, enabling them to adapt interactively to dynamic contexts [4, 17]. Practical examples include autonomous vehicles [4], 3D printers [10] and traffic management solutions [3] that adapt to changing environments - illustrating the adaptability of cyber-physical systems and their impact in different industries [3, 17]. By design, they learn, self-reconfigure and cooperate with other systems [3]. These capabilities arise from the interaction of sensors and algorithms that automate complex processes and at the same time continuously process large volumes of data in order to make well-founded decisions [10, 17]. Technological advancements like miniaturized circuits, faster networks, cost-effective innovations, and semantic web technologies strengthen the intelligence and ubiquity of CPS [3]. However, CPS face several challenges - including their black-box behavior, which makes traceability and transparency difficult [3]. Further challenges arise from contextual influences such as ambient temperature or time, which can trigger anomalies and make system analysis more difficult - especially in view of the high speed and variety of log data in CPS [10].

Explainable cyber-physical systems (ExpCPS) [2] have been developed to close existing transparency gaps. These systems integrate explainability mechanisms, such as ontology-based models, causal reasoning and knowledge graphs, directly into existing CPS infrastructures. By explicitly mapping the relationships between system components and environmental influences, ExpCPS are intended to make it easier to understand why unexpected events occur. This increases traceability and strengthens trust in the system [2].

With the increasing complexity of CPS, it is becoming increasingly important to ensure transparency and traceability. This thesis addresses these challenges by using the SENSE framework to improve explainability through a prompt-based approach.

## 2.2 Semantic Technologies

Semantic technologies offer a structured vocabulary for defining relationships between heterogeneous system components [17]. Ontologies form the basis

for this by describing the central terms of a specialist area and their interrelationships. Knowledge graphs (KGs) can be created on this basis, which are continuously expanded to include causal relationships - for example from log data or context information [10]. KGs are data structures that capture and map real knowledge [9]. Nodes in the graph represent entities, while edges describe their relationships [9]. They enable a structured representation of domain knowledge and support, among other things, conclusions, semantic interoperability and the integration of different data sources [12]. KGs can also be flexibly expanded as they are adaptable to new data, which makes them particularly scalable [9]. However, building such graphs is complex and requires extensive expertise to precisely define entities, relationships and hierarchies. This makes the process both time-consuming and labor-intensive [12].

Semantic technologies - especially knowledge graphs - offer a structured and scalable way of mapping relationships in CPS. However, their implementation often represents a high barrier to entry. This work tackles exactly this problem: It presents a prompt-based approach that supports and simplifies the filling of knowledge graphs through guided interactions with LLMs.

## 2.3 Knowledge Elicitation

Knowledge elicitation refers to the process of capturing knowledge from individuals in order to make it accessible to a larger group of people - for example in organizational contexts [6]. The aim is to use methods and tools that enable knowledge to be collected and reviewed effectively [6]. However, this process is associated with numerous challenges. A key reason is that a large proportion of knowledge is implicit - i.e. deeply rooted in personal experiences and difficult to put into words or formalize [6]. In addition, the ability to clearly formulate knowledge varies greatly between experts, which makes it even more difficult to collect [6]. Moreover, experts can be subject to cognitive biases, such as overestimated self-confidence or the anchor effect. Such effects can lead to the recorded knowledge being inaccurate or distorted [15, 11]. If knowledge is collected from several experts, the complexity increases further: the various contributions must be brought together, taking into account dependencies and correlations [15]. Many knowledge elicitation processes are also unsystematic. They are based on ad-hoc methods, which often leads to inconsistent or unreliable results [11]. There is also often a lack of transparency and comprehensible documentation - which makes it difficult to trace statements or methods used [11]. Effective knowledge elicitation therefore requires close, repeated collaboration with experts, which makes it time-consuming and resource-intensive. [11].

This thesis presents a prompt-based approach that utilizes LLMs and natural language processing methods to specifically support and simplify the process of knowledge elicitation.

## 2.4 LLMs in Knowledge Elicitation

LLMs have developed considerably in recent years and are now able not only to process natural language, but also to link it logically and communicate precisely. This opens up a wide range of possible applications in professional contexts [20]. Among other things, they provide support in the generation of content, the summarization of information and the targeted retrieval of domain-specific knowledge [20]. A key success factor is the ability of these models to adapt to the individual needs and prior knowledge of users [20]. Context-dependent answers that take into account the level of knowledge of the respective person can significantly increase the comprehensibility and relevance of the information and thus increase the practical benefits for stakeholders [20]. A user-centered approach is essential for successful use in professional fields of application [1]. This includes transparent communication of the capabilities and limitations of LLMs so that users can make informed decisions about how to use the model appropriately [1].

Models such as LLaMA or GPT have proven to be particularly promising for complex tasks such as knowledge elicitation. This thesis explores how structured prompt-based interactions can enhance stakeholder understanding and efficiency in knowledge graph construction within the SENSE framework.

## 2.5 Prompt Engineering

Prompt engineering emerges as a systematic process to enhance interactions with LLMs, offering reusable techniques [7] and patterns that improve prompt efficiency and quality [21]. A prompt is a text-based input that is given to an LLM to control its behavior and specifically influence the desired result [14]. It acts as a kind of instruction that provides the model with the necessary context to perform a specific task [14]. A well-designed prompt usually contains a clear instruction for action and can also contain input data, background information or information on the desired output format. [14].

White et al. [21] introduced prompt patterns, which are divided into five central categories, each supporting specific objectives and helping to achieve consistent results:

- **Input Semantics:** focuses on clarity, context and ambiguity resolution

so that LLMs can accurately interpret prompts in different use cases.

- **Output Customization:** allows users to customize the format, tone or audience of responses so that results are tailored to professional or creative requirements.
- **Error Identification and Correction:** supports an iterative improvement process where users incorporate feedback, review details and incrementally refine outputs to increase their reliability.
- **Prompt Improvement:** aims to optimize weak prompts - for example, by breaking complex queries into smaller units or adding examples - to encourage more targeted model responses.
- **Interaction:** emphasizes posing questions rather than merely generating output [21].

Prompt engineering encompasses a variety of techniques aimed at overcoming the challenges of creating effective prompts. The right technique can help to target an LLM and generate relevant, task-appropriate output [14]. In order to maximize our prompting strategy we use instructive prompts [7], directing the model with clear and task-specific instructions. We use a persona technique [14] to dedicate the LLM into an Assistant role. Another technique we use is question-answer prompts [7, 14], where we frame the prompts around specific questions. Not only do we ask the model specific questions, but define questions that the model should ask the user. It is important to follow certain prompting techniques in order to avoid typical pitfalls [7]. Prompts by inexperienced users tend to be vague and lack specificity which lead to unclear, generalized or incomplete responses [7]. Prompts that are formulated too specifically, on the other hand, can limit the flexibility of the model [7].

It is crucial to find a balance between input that is too vague and input that is too detailed. If this balance can be achieved, answers can be generated that can react both precisely and flexibly to different situations. Prompt engineering creates the necessary framework for structured interaction with LLMs, making them usable for specialized tasks. This thesis uses prompt engineering techniques to specifically support LLMs in understanding and responding to user input in the process of implementing the SENSE guideline. Drawing on the work of White et al. [21], which categorizes prompt patterns into five central groups, this thesis integrates selected patterns from each category into the prompt design to increase its effectiveness and ensure consistent, structured output.

## 2.6 User-Centered Design

In order to maximize user centered design, UCD (User centered design) principles were integrated into the development of the prompt. UCD emphasizes iterative system development centered on user needs [19]. Through repeated cycles of prototyping and testing, UCD practitioners can align functionality with user expectations and ensure products remain useful [13]. Methods such as task analysis make it possible to observe users in real application contexts and divide complex workflows into manageable individual steps. In this way, key design aspects can be identified. [13, 19]. An iterative design approach proves particularly helpful in prototyping: stakeholders can provide feedback on user-friendliness, efficiency and accuracy at every stage of development. [19]. Scenario-based evaluation provides additional insights and deepen the understanding of user-relevant requirements [19]. By applying user-centered design methods - especially iterative prototyping and systematic usability evaluations - user satisfaction can be improved in a targeted manner [13]. Open feedback loops and the inclusion of real user experiences help to develop prototypes that meet the specific requirements of a domain [13, 19].

A user-centered design approach ensures that the prompt-based approach meets the expectations and needs of stakeholders and makes complex frameworks such as SENSE more accessible. By integrating UCD principles into the design of the prompt, this work aims to increase user-friendliness and support users specifically in the application of the SENSE guideline.

## 3 Methodology

This section describes the methodological approach of this work, which aims to systematically develop and evaluate a structured prompt to support users in gathering domain knowledge according to the SENSE guideline. To achieve this, a combination of the design science research (DSR) framework and a structured evaluation framework is employed. The DSR framework provides the overarching methodological structure for the iterative development process, while the evaluation framework guides the systematic assessment of the developed prompt based on defined criteria. The evaluation involves a comparative analysis of four LLMs: Deepseek-r1 (70B), LLaMA 3.3 (70B), GPT-4, and GPT-4 Turbo.

Following the presentation of the methodological foundations, the prompt development process is described in detail, highlighting the iterative refinement through practical testing. Subsequently, the specific prompt design and engineering patterns applied during the development phase are introduced.

The technical implementation, including the computational setup and evaluation infrastructure, is then outlined. Finally, a concrete use case is presented that illustrates the practical context in which the developed prompt is to be used. In this section, the structure and requirements of the SENSE guideline are explained in more detail, highlighting in particular the steps that users must follow to document system knowledge in a structured and semantically consistent manner.

### 3.1 Design and Evaluation Framework

This thesis uses DSR framework as the overarching methodological structure guiding the iterative development of the prompt. In addition, a structured evaluation framework is applied to systematically assess the quality and effectiveness of the developed prompt. Together, these frameworks address the practical challenge of supporting users in the application of the SENSE guideline and the integration of the explainability stack into existing systems.

DSR aims at the development and evaluation of artifacts [5] and is therefore particularly well suited to systematically address practical challenges - such as the development of a prompt-based support for the elicitation of domain knowledge. The DSR approach is divided into three interconnected cycles [8]:

- **Relevance Cycle:** The research work is based on the practical challenge of simplifying the creation of KGs for CPS - in compliance with the requirements of the SENSE framework. This requirement forms the basis for the design of the prompt and ensures that the work addresses concrete, practical issues.
- **Design Cycle:** The iterative design and further development of the prompt-based support follows this cycle. The prompt is continuously created, tested and revised in order to meet the needs of users and improve the performance of the system.
- **Rigor Cycle:** The research draws on existing theoretical knowledge to design an artifact that is both innovative and based on proven methods and findings [8].

To complement the DSR approach, an evaluation framework was developed, specifically geared towards the systematic assessment of the prompt. The evaluation focuses on three main criteria: *completeness* (whether all elements of the prompt are addressed), *relevance* (whether the output is on topic and appropriate in context), and *adherence to instructions and patterns*

(whether the output follows certain rules and patterns). To perform the evaluation, a comparative analysis was conducted using four LLMs: Deepseek-r1 (70B), LLaMA 3.3 (70B), GPT-4, and GPT-4 Turbo. These models were evaluated based on the outputs generated when interacting with the developed prompt, following a structured testing procedure.

The evaluation results provide valuable feedback that is integrated into the iterative design process, contributing to the gradual improvement of the quality, usability, and effectiveness of the prompt as part of the design science research cycle.

The next section will take a look at the development process of the prompt. This process was shaped by the design cycle of the DSR framework and focused on iterative refinement through practical testing. The section explains how the prompt was gradually improved over multiple versions to better support users in applying the SENSE guideline.

### 3.2 Prompt Development Process

A structured prompt was developed to help users apply the SENSE guideline step by step and to facilitate the integration of the explainability stack into existing systems. The core of the approach is a carefully formulated prompt that triggers a step-by-step, dialog-based interaction with a LLM. The prompt guides users through the entire process by asking specific questions, providing contextual explanations, and ensuring that individual steps are completed in the correct order. This design aims to make the complex knowledge elicitation process accessible and understandable, particularly for users without prior experience in semantic technologies.

The development of the prompt followed an iterative process aligned with the design cycle and comprised four versions. Each version was refined through practical testing of the model’s behavior, with the goal of improving the flow of the conversation, minimizing misunderstandings, and ensuring that the LLM consistently adheres to the SENSE guideline. The final version focuses on the first three steps of the guideline: establishing a common understanding of terms, clarifying system objectives, and identifying key system components and their interrelationships in a structured and comprehensible manner.

After outlining the prompt development process, the focus now shifts to the design and engineering patterns that structure and govern the interaction flow.



### 3.3 Prompt Design and Engineering Patterns

The prompt is designed as a guided interaction and supports users in integrating the SENSE explainability stack into their system. In the role of an interactive assistant, it ensures that integration takes place along the SENSE User Guideline - precisely and consistently across all steps. To begin with, the prompt creates a common conceptual foundation by defining central system components such as platforms, sensors and hierarchical relationships. These terms form the framework of the interaction and must be consistently observed throughout the entire process.

To ensure a structured and reliable interaction, the prompt follows **three central rules of behavior**. First, it contains a **fact-checking** mechanism: After each major step (step 1, 2 and 3), the LLM creates a list of critical assumptions that need to be checked by the users. Second, a **reflection** step is integrated, where the model reveals its reasoning after each answer and makes assumptions transparent. Thirdly, all data must be output in a structured, tab-delimited **table format** (CSV) so that it can be transferred directly into the provided SENSE Excel template that accompanies the guideline. This multi-sheet template, for example, provides predefined columns such as *Platform*, *PlatformType*, *hostedBy\_Platform*, and so on; with a simple copy-and-paste of the LLM-generated CSV tables, the worksheets are populated automatically.

The prompt is designed as a step-by-step (step 1, 2 and 3 including sub-steps 3.1 and 3.2) process that systematically gathers and validates system information. In step 1, the LLM introduces itself, explains key terminologies and outlines the behavioral rules before moving forward. In step 2, the user defines goals of integrating the stack into their system and identifies key questions that need to be addressed. The model then breaks down these responses into structured components (and sub-questions, if needed) and verifies them for accuracy. Step 3 focuses on building a representation of the system, guiding the user in listing all platforms, sensors and their relationships. In steps 3.1 and 3.2, the user further specifies platform types, sensor types, and observable properties, while the LLM ensures consistency in hierarchical connections and overall data structure.

To make it easier to understand, the prompt uses a simplified example ("toy example") in which a household platform with a charging station for electric vehicles (EV charger) and a battery is introduced. This example is gradually expanded over the course of the interaction and serves to illustrate how platforms, sensors and their relationships should be documented correctly. Through clear explanations of terms, structured validation steps and a systematic process, the prompt ensures that users record their sys-

tem completely, correctly and consistently in accordance with the SENSE guideline.

We used seven prompting patterns based on the paper *"A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT"* by White et al.[21]. For a concise overview of these patterns and their main functions, see Table 1. In the following subsections, each pattern is briefly introduced and illustrated within the context of our prompting strategy. The prompts are written in a first-person perspective, addressing the user directly by using the pronoun *me*. The full prompt can be viewed in Appendix A.

Pattern Name	Description
Meta Language Creation	Sets a unified vocabulary to ensure clarity.
Flipped Interaction	Guides the LLM to ask questions, enabling a user-driven dialogue.
Persona	Assigns a fixed role to maintain consistent tone and style.
Cognitive Verifier	Splits inputs into smaller parts and checks each for accuracy.
Fact Check List	Provides verifiable statements for user confirmation.
Reflection	Reveals the model’s assumptions to foster transparency.
Template	Enforces a standard output format (CSV).

Table 1: Overview of the seven Prompt Engineering Patterns utilized

### 3.3.1 Meta Language Creation Pattern

When integrating the SENSE explainability stack it is crucial that both the user and the LLM share a clear and consistent terminology. This pattern was chosen to establish a "meta-language" that removes ambiguity about the meaning of key terms (e.g. "Platform", "Sensor" or "hostedBy"). By implementing this pattern we introduce a concise set of definitions everyone must follow. In addition, the pattern streamlines interactions by allowing symbolic notations that prevent confusion or misinterpretation of specialized terms.

First, we introduce the LLM to the terminology used in the SENSE guideline by declaring a "meta-language":

```
Let's define a concise "meta-language" for clarity.
```

Whenever the following terms are used, they carry the meanings below:

Subsequently, we define terms like platform, sensor or hostedBy:

#### **Platform**

A Platform is any device, facility or logical group that can host sensors measuring data.

Devices, facilities and logical groupings are all considered Platforms in the SENSE stack.

#### **Sensor**

A Sensor measures some property in the system and is always hosted by a Platform.

Each sensor is physically or logically located on exactly one Platform.

#### **hostedBy**

A hierarchical relationship indicating that a sub-platform (or a sensor) is hosted by a higher-level Platform.

In the SENSE stack, the first (top-level) Platform is typically considered the "host" and any nested Platforms or Sensors are said to be hostedBy it.

### **3.3.2 Flipped Interaction Pattern**

Following the process of implementing the SENSE guideline, the user must provide detailed information about their system. Therefore, this process involves a lot of data gathering and knowledge elicitation. This pattern was selected to allow the LLM to lead the questioning process rather than passively waiting for user input. The LLM is intended to ask predefined questions to close the knowledge gaps and is encouraged to pose follow-up

questions if necessary, refining or adjusting details with each iteration.

The process begins with collecting information on the user's system. At this stage, the user must define their goals and state the main problem they aim to address:

```
I want you to ask me questions so we can collect all
relevant information step by step.
Specifically :

Which questions should be answered by the system?

What are important anomalies you want to detect and
explain?

What is the purpose of doing this?

What is the end result we expect?

What are the limitations and scope?
```

Once the goals have been defined, the LLM prompts the user to list all platforms, sensors and connections between platforms as well as connections between platforms and sensors:

```
Ask me to list all Platforms in my system.

Ask me for the hostedBy connections between these
Platforms.

Ask me to list all Sensors in my system.

Ask me to list all connections between Sensors and
Platforms (to map each sensor to its relevant
platform)
```

Next, the user is asked to specify the types of platforms and sensors, along with the observable properties measured within the system:

```
Ask me to define the types of the Platforms, the
Sensor types, and the observable properties they
measure.
```

To further refine the LLM’s understanding of the existing system, the user is asked to provide concrete instances of each platform and sensor, including their hierarchical connection and observable properties measured by the sensors within the system:

```
Ask me for the actual instances of each Platform and
Sensor .

Ask how they connect hierarchically (using hostedBy) .

Ask me to confirm which observable property each
Sensor measures .
```

### 3.3.3 Persona Pattern

This pattern is used to place an LLM into the roles of a dedicated expert assistant. It not only adjusts the model’s tone but also ensures that its responses remain consistent with the responsibilities of the defined persona. By applying this pattern, the LLM is more likely to generate responses as if it were the expert itself, referencing the appropriate terminology.

The model is explicitly introduced to its role in integrating the SENSE stack. It is instructed to always maintain its given persona of a dedicated assistant throughout the interaction:

```
You will guide me through integrating the "SENSE
Explainability Stack" into my system .
Always maintain the persona of a dedicated assistant
focused on accurately capturing the system's
structure and causal/state relationships , ensuring
that I am supported when I enter the data that
describe my system .
```

### 3.3.4 Cognitive Verifier Pattern

Complex questions related to sensor data, platform hierarchies or causality can benefit from a systematic breakdown. This pattern is designed to ensure that the LLM systematically verifies each response for correctness. By applying this prompting pattern, we aim to improve accuracy by checking smaller segments of information individually. Additionally, the LLM is encouraged to synthesize a more comprehensive final answer after verifying all

segments.

Whenever the user provides input about their goals in step 2, the LLM is instructed to break the answers down into sub-questions if needed, confirm each element and synthesize the information into a structured final output:

```
When you see my answers , please :  
  
1. Break them down into smaller sub-questions or  
points if needed for clarity .  
  
2. Confirm each sub-point to ensure there are no  
misunderstandings .  
  
3. Synthesize the final structured information  
(platforms , sensors , connections , etc.) before moving  
on .
```

### 3.3.5 Fact Check List Pattern

Factual accuracy is essential when populating a knowledge graph. This pattern instructs the LLM to list verifiable facts, making it easier for humans to confirm or refute them. The goal is to foster a habit of self-auditing, allowing the user to identify statements that may require external verification.

By implementing this pattern, we aim to prevent silent errors or assumptions from slipping through. After each answer, the LLM is instructed to append a fact check list - defined in the prompt as a behavioral rule - so that users can independently review and validate the most critical facts.

The implementation process of the SENSE guideline is divided into multiple steps (steps 1, 2 and 3) representing major sections. After one section (step) is completed, the LLM is instructed to fact check critical statements or assumptions:

```
After each major section (Step 1, Step 2, etc.) ,  
please provide a fact check list of critical  
statements or assumptions I've made. Label them as :
```

```
Fact to Check: "X."  
Potential Source or Reason: "Why we need to verify it  
."  
Possible Consequence: "What might happen if it's
```

incorrect."

For example:

Fact to Check: "The EV Charger can always cause a peak demand."

Potential Source: I observed large energy spikes in the past.

Possible Consequence: If untrue, we might overestimate the battery's impact.

This helps ensure accuracy before we lock in any detail.

### 3.3.6 Reflection Pattern

This pattern encourages the LLM to articulate its reasoning in order to increase transparency and make it easier to detect misunderstandings. By applying this pattern, we aim to highlight possible leaps in logic or unsupported inferences. This pattern represents one of the three behavioral rules.

After providing a response, the LLM is instructed to reflect and explain why it arrived at a specific conclusion:

After you provide any answer or summary, add a short Reflection section:

Explain why you arrived at that specific conclusion or structure.

Clarify any assumptions or inferences you made.

Example:

Reflection:

I noticed you mentioned a battery but didn't confirm its type. I assumed it was a standard Lithium-Ion battery

because of the mention of 'StateOfChargeSensor'. If that assumption is incorrect, let me know, and I'll adjust accordingly.

This helps me see your reasoning process and correct any misunderstandings.

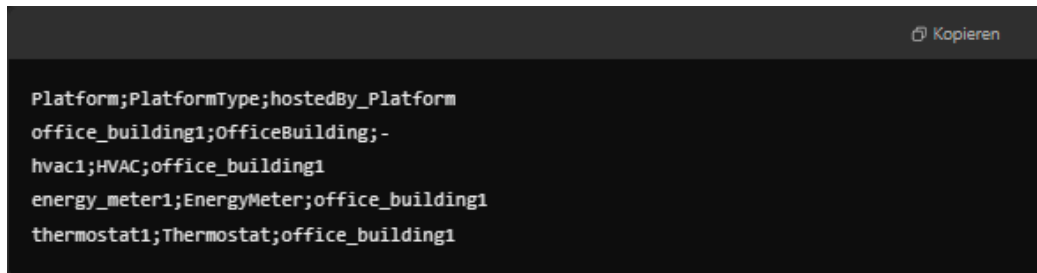
### 3.3.7 Template Pattern

In order to integrate the SENSE explainability stack, the collected information about platforms, sensors, states and related elements has to be transferred into a predefined Excel template. This pattern ensures consistent formatting with the goal to reduce effort when exporting data into external tools such as spreadsheets. To support this, the LLM is instructed to follow a fixed syntax, being defined as the third behavioral rule.

Any collected system information that needs to be displayed as a table should be formatted using the following syntax:

```
Header1 ; Header2  
Value1 ; Value2  
...
```

The following example 1 illustrates how the LLM outputs user-provided data:

A screenshot of a dark-themed interface showing the output of an LLM. The output is a CSV-style table with three columns: Platform, PlatformType, and hostedBy\_Platform. The first row is a header, and the following four rows contain data for different sensors in an office building. A 'Kopieren' button is visible in the top right corner of the interface.

```
Platform;PlatformType;hostedBy_Platform  
office_building1;OfficeBuilding;-  
hvac1;HVAC;office_building1  
energy_meter1;EnergyMeter;office_building1  
thermostat1;Thermostat;office_building1
```

Figure 1: Example output of the LLM in CSV-style table format.

After establishing the conceptual foundation through the prompt design patterns, we proceed to the implementation layer that enables their practical execution.

## 3.4 Technical Implementation

To ensure the reproducibility and reliability of the model evaluations, a detailed technical setup was established. This section outlines the server environment, containerization approach, and the procedures used to deploy and manage the LLMs during the evaluation process.

LLaMA 3.3 (70B) and Deepseek-r1 (70B) were run on a GPU-enabled virtual server provided by the Vienna University of Economics and Business. GPT-4 and GPT-4 Turbo were used via the ChatGPT web interface. The



server runs under Ubuntu 22.04.5 LTS with the kernel 6.8.8-4-pve and is accessible via SSH as long as a connection to the university's VPN is established. The server infrastructure includes a storage capacity of 250 GB, 16 GB RAM, two NVIDIA RTX A5000 GPUs (each with 24 GB memory, 48 GB GPU memory in total) and a 4-core CPU. The system is based on the x86\_64 architecture. The NVIDIA CUDA Toolkit (version 12.4), which provides the required drivers and libraries, to optimize the use of GPU resources. The NVIDIA System Management Interface (nvidia-smi) tool was used to monitor GPU performance - for example in terms of memory utilization, temperature or load. The container environment was set up using the Docker Engine (Community Edition, version 27.5.1). Plugins such as buildx and compose were available to support the container configuration. The nvidia runtime environment was used for GPU acceleration within the containers. A pre-configured container image (*ollama/ollama*) served as the basis for managing and running any LLM. To ensure data persistence, a Docker volume was mounted to store files under `/root/.ollama/models`. This meant that access to the stored data was retained even after the container was restarted or stopped. To interact with the models via an API, port 11434 was forwarded from the container to the host system.

In the following we are going to showcase example shell commands when interacting with the server. In order to start a container, we used this command:

```
docker run -d --gpus all -p 11434:11434 --name  
<container_name> -v <volume_name>:/root/.ollama/  
models <image_name>
```

To pull the LLaMA and Deepseek model from <https://ollama.com/library> we used this command:

```
docker exec -it <container_name> ollama pull  
<model_name>
```

In order to run the pulled model, we used this command:

```
docker exec -it <container_name> ollama run  
<model_name>
```

In the following, an explanation of the terms used can be seen:

```
-d: Detached mode starts the container in the  
background, so the terminal doesn't get blocked.
```

```
--gpus all: Activates GPU support for the container ,  
allowing access to all available GPUs.  
-p: Port mapping from the host to the container .  
--name: Assigns a custom name to the container for  
easier identification .  
-v: Volume mapping connects a host directory to a  
directory in the container .  
-it: Enables interaction with the container .
```

Having established the technical environment, the next section introduces a representative use case to illustrate the practical application of the prompt and its alignment with the SENSE guideline.

### 3.5 Use Case

In order to demonstrate the practical applicability of the developed prompt and its alignment with the SENSE guideline, a representative use case is defined. This involves a user who has the task of integrating the explainability stack into an existing system using the structured approach of the SENSE guideline.

Although the guideline provides a comprehensive methodology for converting raw system information into a knowledge-based representation, its technical complexity and the use of domain-specific terminology make it difficult to implement correctly, especially for users without a background in knowledge engineering.

The SENSE guideline supports the integration process by guiding users through five clearly structured steps, thus enabling the connection of the existing system to the SENSE explainability stack - a prerequisite for semantic-based system explanations. During the process, users capture system-relevant knowledge in a provided Excel template that is fully compatible with the architecture of the SENSE Core. This template contains various tabs and columns that must be filled in step by step to ensure standardized and complete documentation.

In the context of this thesis, the focus is on the first three steps of the guideline, which are crucial for building a common conceptual foundation:

- **Step 1:** Common conceptualization.

The SENSE guideline defines a domain-specific vocabulary for modeling the system. This includes capturing platforms, sensors and their relationships, especially the "hostedBy" relationships that represent hierarchical structures.

- **Step 2:** Understanding goals and user needs.

In this step, the main goals and expectations of the system behavior are specified. Users define which questions the system should answer and which anomalies it should recognize and explain. These goals serve as a guideline for further modeling decisions.

- **Step 3:** Identify system components.

Users document the concrete system components - platforms, sensors, observable properties and their hierarchical relationships. This is where the transition from abstract concepts to concrete instances in your own system takes place.

The challenge is to apply these steps correctly, maintain consistent terminology, and ensure structured documentation. The developed prompt meets these challenges through targeted questions, contextual explanations, and structured guidance through the interaction with the LLM. It helps to keep the focus on the defined objectives and supports the systematic filling of the Excel template in accordance with the requirements of the SENSE guideline.

This use case illustrates how the developed prompt facilitates the practical implementation of the SENSE guideline - especially for users without in-depth technical knowledge. By providing targeted support in the critical initial steps, the prompt aims to lower the barriers to entry and helps to successfully integrate the explainability stack into existing systems.

## 4 Evaluation and Results

This section presents the evaluation of the developed prompt and the comparative analysis of the output generated by four different LLMs. The aim of the evaluation is to systematically examine the extent to which the prompt supports the application of the SENSE guideline. The focus is on the criteria *completeness*, *relevance* and *adherence to instructions and patterns* in the model responses.

The evaluation process is structured as follows: Section 4.1 introduces the evaluation criteria and the scoring system used to assess the model outputs. Section 4.2 describes the test scenario developed to standardize model interactions. Section 4.3 provides a detailed assessment of each model's performance based on the defined criteria. Section 4.4 summarizes the findings through a comparative analysis, highlighting key differences between the models.

This evaluation provides important insights into the strengths and weaknesses of different LLMs in supporting knowledge elicitation tasks and the practical application of the SENSE framework.

#### 4.1 Evaluation Criteria and Scoring System

The output of each LLM is evaluated based on three key criteria: *completeness*, *relevance*, and *adherence to instructions and patterns*. Each output is assigned a score from one to five, with five representing the highest level of performance. The chosen evaluation framework is designed to ensure an objective and consistent assessment of each model’s ability to support the implementation of the SENSE guideline.

The *completeness* criterion checks whether the model’s response fully covers the required topics of the prompt. This includes all required topics, questions and components - without omissions. A checklist was created for the assessment, which is based on the structure of Step 1, Step 2 and Step 3 of the SENSE guideline. The outputs of the models were compared with this checklist to verify that all required content was presented in a structured and coherent manner.

Score	Description
1	Several main steps (Steps 1, 2 and 3) largely incomplete or entirely missing. The majority of questions unanswered or answered superficially.
2	At least one entire main step (e.g., Step 3) substantially incomplete or major components missing. Multiple details or answers clearly missing or insufficient.
3	All main steps (1-3) addressed, but at least one sub-step (3.1 or 3.2) incomplete or partially omitted. At least one question from Step 2 clearly incomplete or vague.
4	All main steps (1-3) covered clearly; exactly one sub-step (3.1 or 3.2) superficially addressed or with minor missing details. No critical or significant elements omitted.
5	All steps (Step 1, Step 2, Step 3 incl. 3.1 and 3.2) explicitly and fully covered. No questions or required details missing. All definitions and instructions explicitly addressed.

Table 2: Scoring Rubric for Completeness

The *relevance* criterion assesses how closely the model’s answers match the objective and content framework of the prompt. The output should clearly focus on the task at hand and not introduce extraneous or superfluous knowledge. To assess this, the responses were compared with the intended purpose and context of the prompt. The aim was to identify potentially distracting or unnecessarily lengthy passages that could affect the clarity of the interaction.

Score	Description
1	Majority of responses clearly irrelevant or off-topic. Core objective heavily obscured or not recognizable.
2	Noticeable portions of the responses irrelevant or significantly off-topic, affecting clarity. Core objective still recognizable but noticeably impaired.
3	Responses predominantly relevant, but contain two or more clearly identifiable minor irrelevant elements that slightly distract from overall understanding.
4	All responses relevant; maximum one brief instance of slightly off-topic information present, not affecting overall clarity.
5	All responses directly relevant and fully aligned with the prompt. No off-topic information present.

Table 3: Scoring Rubric for Relevance

The *adherence* criterion examines the extent to which the model takes into account the specific instructions and predefined prompting patterns. This includes the three defined behavioral rules: the inclusion of fact check lists, reflection sections and compliance with the required table format (CSV structure). A list of all expected behaviors and structural elements was created. The outputs were checked to ensure that these requirements were implemented consistently and correctly throughout the interaction.

Score	Description
1	Behavioral rules mostly ignored or not implemented at all. Persona role absent, ignored, or fundamentally misunderstood.
2	Multiple behavioral rules significantly neglected, incomplete, or clearly incorrect. Persona role heavily inconsistent or significantly neglected in several responses.
3	At least one behavioral rule regularly incomplete or superficially implemented in multiple responses. Persona role recognizable but noticeably inconsistent or fluctuating between responses.
4	All behavioral rules generally implemented correctly, but minor inconsistencies or single isolated deviation observed in exactly one rule. Persona role maintained clearly, with only slight deviations.
5	All three behavioral rules (Fact Check, Reflection, CSV-format) precisely and consistently implemented throughout. Persona role consistently maintained in every response. All given instructions explicitly followed without exception.

Table 4: Scoring Rubric for Adherence to Instructions and Patterns

To apply the defined evaluation criteria in a controlled and consistent manner, a test scenario was developed that reflects a realistic system setting. The scenario serves as a common basis for generating comparable outputs across all evaluated models.

## 4.2 Test Scenario

To ensure a standardized and fair evaluation of the developed prompt across different language models, a representative test scenario was designed. This scenario depicts a realistic system environment and focuses on monitoring energy consumption and the efficiency of climate control in an office building. It provides a consistent context to systematically evaluate the *completeness*, *relevance* and *adherence* of the model responses with respect to the requirements of the SENSE guideline.

All models except for GPT-4 Turbo were evaluated based on the following test scenario. The chosen scenario is about an office building’s energy consumption and climate control efficiency, where the goal is to monitor anomalies related to unexpected high energy consumption or inefficient tem-

perature regulation. The system should not only detect irregularities but also provide explanations for their root causes.

To structure this scenario, we defined a set of platforms and sensors that reflect a real-world setup. The platform hierarchy consists of an **office building**, which hosts several key platforms: an **HVAC system**, an **energy meter**, and a **thermostat**. Each of these platforms is assigned a type, with **office\_building** as the main entity, **hvac\_system** responsible for temperature control, **energy\_meter** for tracking electricity usage, and **thermostat** for localized temperature regulation.

The system also includes sensors and their assignments, which are critical for monitoring and analyzing data. A **temperature sensor** is placed on the **thermostat** to measure indoor climate conditions, an **active power sensor** is attached to the **energy meter** to track electricity consumption, and a **state sensor** monitors the operational status of the **HVAC system**. These sensors are explicitly linked to their respective platforms and assigned specific observable properties, such as **temperature**, **active power** and **HVAC state**.

GPT-4 Turbo was evaluated on a similar test scenario that has been provided by a user during the prompt testing phase. While the scenarios vary in terms of platforms, sensors, their relationships, and observable properties, the core setup - both involving a building environment and energy consumption anomalies - is largely consistent.

### 4.3 Model Evaluation

In the following, each model is evaluated individually and receives a score from one to five for each criterion. The evaluation is structured in the same way for each model: First, the respective criterion is stated, followed by the score awarded and a justification for the rating. This is followed by a detailed description of the interaction with the model, which illustrates how the model performed with regard to the respective criterion. To support transparency and traceability, Appendix A includes all LLM outputs used for evaluation - shown as screenshots for all models except GPT-4 Turbo, which is accessible via a direct chat link.

#### 4.3.1 Deepseek-r1 (70B)

##### Criterion: Completeness

**Score 1:** several main steps - particularly step 1 and step 2 - were either omitted or superficially addressed. The model failed to clarify its role,

skipped user interaction and provided incomplete or generic answers instead of following the structured prompt.

**Prompt Element: Step 1 - Introduction and Role Clarification**

The first step was not addressed at all. The LLM interpreted the input given wrong, assuming it should integrate the SENSE stack into its own system rather than the user's: "To integrate the SENSE Explainability Stack into my smart home system, I will follow a structured approach based on the provided steps and example.". It did not make the user aware of its role and did not provide any context surrounding the guideline. There was no brief overview over the prompt's terminology or definitions that is used in the guideline. The three behavioral rules defined in the prompt were not mentioned, despite the LLM being explicitly instructed to inform the user about them in step 1. Without asking the user whether they were ready to continue, the model proceeded directly to step 2.

**Prompt Element: Step 2 - Goal Definition and Questioning**

Instead of asking the user predefined questions, the model answered them right away, acting as though it were describing its own system. It responded to all the questions listed in the prompt, leaving none unanswered. The model continued with step 3 without checking if the user was ready to proceed.

**Prompt Element: Step 3 - System Representation and Typing**

The model used the toy example to list all platforms, sensors and connections between platforms and sensors. It copied the types of platforms and sensors directly from the example, including actual instances and observable properties. These were also taken from the example, without user input.

After its initial response, the user corrected the model, clarifying its actual role and the system it was meant to support. The model successfully changed the approach to using "your system" instead of "my system". It restarted the process from Step 1, this time introducing the terminology, but not its role or any context. The model did not use any of the predefined terminology definitions. The behavioral rules were once again ignored. The model defined a generic approach section and next steps to follow, directly jumping to step 3 and bypassing the stepwise interaction and required schema entirely.

**Criterion: Relevance**

**Score 2:** noticeable portions of the response - especially in step 2 where the model answered questions that were never asked - were off-topic. While the



core objective remained recognizable, the initial misalignment and unnecessary answers impaired overall clarity.

In prompt element step 2, the model incorrectly answered a set of questions regarding the goals of using the SENSE stack. Every question asked and each corresponding answer given can be rated as irrelevant, as they were never requested in the first place.

After the model was corrected and restarted, it explained the predefined terminology using different wording. Although it did not stick to the correct wording, it used definitions that are in scope of the domain. For example, in the prompt sensors are defined as something that "measures some property in the system", the model defined sensors as "devices that detect events or change their environment and end signals". It is logically correct and therefore considered relevant.

#### **Criterion: Adherence to instructions and patterns**

**Score 1:** none of the required patterns were consistently followed; most were completely ignored or incorrectly applied. The persona only emerged after user correction, therefore not being maintained throughout the interaction.

The fact check pattern: was not implemented at any point. The reflection pattern: was also entirely omitted. The template pattern: was not followed, as tables were not created in CSV format. The persona pattern: was only established after the model was reminded. The meta-language pattern: while terminology was used correctly, definitions were phrased inconsistently throughout the interaction. The flipped interaction pattern: questions were not asked but answered directly by the model itself. The cognitive verifier pattern: no sub-questions were asked in step 2.

#### **4.3.2 LLaMA 3.3 (70B)**

##### **Criterion: Completeness**

**Score 3:** all three main steps were addressed, but step 2 lacked proper user questioning and skipped interaction prompts, while step 3 required user guidance and omitted key details like observable properties, which lead to partially completed sub-steps.

##### **Prompt Element: Step 1 - Introduction and Role Clarification**

The model introduced itself, stating its role as an assistant: "I'm delighted to be your dedicated assistant in integrating the SENSE Explainability Stack into your system. My primary goal is to ensure that we accurately capture the structure and causal/state relationships within your system, providing you with seamless support throughout this process.". The model followed up with giving a brief overview of the terminology used in the prompt and made the user aware of the rules it follows. The model asked the user before continuing with step 2.

#### **Prompt Element: Step 2 - Goal Definition and Questioning**

Instead of asking predefined questions the model offered suggestive answers, the user could choose from: "For example are you looking to: Optimize energy usage? Predict and manage peak demand? Identify causes of inefficiency? Enhance system reliability?". While this approach introduced useful suggestions, it deviated from the expected prompt structure. It then proceeded with elements of step 3 immediately without asking for permission.

#### **Prompt Element: Step 3 - System Representation and Typing**

The model used platform examples from the prompt to illustrate what a platform might be, following up by asking to list the systems types of platforms. After entering the users goals and list of platforms, the model correctly repeated the list. It continued by asking to list all sensors and connections between platforms and sensors using "hostedBy". The user had to explicitly make the model aware of the platform types and sensor types in their system. The model did not ask the user about observable properties in their system. The model did not ask about observable properties of the system, but correctly handled all other required information.

#### **Criterion: Relevance**

**Score 3:** while the model's responses were largely relevant, they included several minor off-topic elements, such as suggesting next steps and adding unnecessary purposes, which slightly distracted from the prompt's intended structure and focus.

The model did not ask the predefined questions from step 2, but instead offered possible answers. While it failed to ask predefined questions, the offered answers were still somewhat relevant to the overall task. When addressing system sensors, the model proposed possible sensor types instead of prompting the user, which was not expected but still related to the topic. When discussing observable properties, the model suggested potential properties

rather than asking about them, which does not align with the instructions, but loosely relevant. The LLM mentioned various purposes of the properties, which went beyond the prompt’s scope. After completing steps one to three, the model proposed possible next steps, which was not asked for. This is irrelevant for the purpose of this prompt.

**Criterion: Adherence to instructions and patterns**

**Score 2:** several patterns, such as fact check, reflection and flipped interaction were only partially or inconsistently applied, while other patterns like persona and meta-language were followed more reliably.

The fact check pattern: was used until the point when the user asked about observable properties which have not been discussed until this point. The reflection pattern: was also followed until this same point, but was not applied beyond it. The template pattern: was incorrectly applied, not presenting data in CSV-style format. The persona pattern: was used throughout the entire interaction. The meta-language pattern: was consistently maintained throughout the entire interaction. The flipped interaction pattern: the model did not ask the predefined questions in step 2, but managed to ask all questions from step 3, not covering 3.1 or 3.2. The cognitive verifier pattern: no sub-questions were asked in step 2.

### 4.3.3 GPT-4

**Criterion: Completeness**

**Score 3:** all main steps were clearly addressed. Minor omissions in sub-step 3.1 and 3.2, such as not explicitly asking for sensor or platform types and observable properties, which none of critically impacted the completeness of the responses.

**Prompt Element: Step 1 - Introduction and Role Clarification**

The model successfully introduced itself and its role to the user. It explained relevant terminology and provided an overview of the three behavioral rules. The user was asked for confirmation before proceeding to step 2.

**Prompt Element: Step 2 - Goal Definition and Questioning**

The model asked eight questions, none of which were directly taken from the predefined list, but they were similar in content and meaning. The user was provided with an example to support the understanding of the topic.

Therefore, the model applied the toy example correctly, which was provided in the prompt. The given answers were grouped into relevant categories, such as "Specific Anomalies to Detect and Explain" or "Scope and Limitations". The user was asked for confirmation before proceeding with the next steps, not specifically mentioning step 3.

**Prompt Element: Step 3 - System Representation and Typing**

The model asked to list all platforms, sensors and hostedBy connections. Additionally, the model provided an example data entry to help the user understand how to structure their response. Although the model did not ask to list platform or sensor types, the example data entry displayed that such specifications were necessary. The model correctly showed all information given in a table format. Observable properties were not explicitly requested, but the model assumed their relevance and prompted the user to confirm or revise this assumption.

**Criterion: Relevance**

**Score 4:** the model's responses were consistently focused and aligned with the task, with only one minor deviation. Categorizing the user's goal did not affect overall clarity, but was not explicitly required.

The model grouped the user's answers related to their goals, which was not explicitly asked for, yet remained contextually relevant. Overall, the model's behavior throughout the interaction demonstrated a high level of relevance to the prompt.

**Criterion: Adherence to instructions and patterns**

**Score 2:** several patterns, such as fact check, reflection and flipped interaction were only partially or inconsistently applied, while other patterns like persona, template and meta-language were used correctly.

The fact check pattern: was not done for step 1 or 2 but was used for later responses. It was followed correctly after the user provided their goals and again after platform and sensor lists were entered, including during the correction of observable properties. The reflection pattern: was not used in step 1, but was applied consistently in all following responses. The template pattern: was followed correctly, with tables created in CSV format. The persona pattern: was maintained consistently throughout the entire interaction. The meta-language pattern: was followed, with terminology and definitions used correctly. The flipped interaction pattern: predefined questions were not

asked; instead similar and additional questions were used. It also omitted questions regarding platform or sensor types and observable properties. The cognitive verifier pattern: no sub-questions were asked in step 2.

#### 4.3.4 GPT-4 Turbo

##### **Criterion: Completeness**

**Score 5:** all components of the prompt are thoroughly addressed with detailed and comprehensive coverage.

##### **Prompt Element: Step 1 - Introduction and Role Clarification**

The model introduced itself as a dedicated assistant whose role is to help define and structure the user's system, ensure accurate relationships between components as well as guide the correcting formatting of data entries (CSV-style). Key terminology was appropriately used and the three behavioral rules were clearly explained, providing examples of the rules to support understanding. The LLM asked before proceeding with step 2.

##### **Prompt Element: Step 2 - Goal Definition and Questioning**

The model accurately asked the five predefined questions and added sub-questions where appropriate. After the user provided initial answers, the LLM repeated all answers and subsequently began to clarify and refine the answers given. The model emphasized reviewing the previous summary to ensure full understanding before proceeding with step 3. The user confirmed the model's summary was correct and addressed the model's follow-up questions in more detail. The LLM revised and refined the summary again to reflect all aspects. After confirming that all information given was correct, the model continued with step 3.

##### **Prompt Element: Step 3 - System Representation and Typing**

The model asked the user to map out their system by inputting system representations such as platforms, sensors and connections. The LLM instructed the user to provide any data in a CSV format and supplemented this with relevant examples. After receiving the system data, the model summarized the input and continued with step 3.1, addressing hostedBy connections and observable properties. After providing the users data, the model again summarized the data given and continued with step 3.2, assigning each platform and sensor to its type and confirm their hierarchical relationships. The user verified the input.

##### **Criterion: Relevance**

**Score 2:** while the majority of the response was closely aligned with the prompt, the introduction of an invented step 4 - including detailed, fabricated content - represents a significant off-topic deviation.

The model remained on-topic throughout steps 1 to 3 and did not introduce any irrelevant content within the defined scope of the prompt. However, the LLM incorrectly attempted to continue with step 4, which was not part of the initial prompt and therefore non-existent. The user agreed to continue with step 4, not being aware that the step was not defined. The model proposed defining relationships between variables, identify causal dependencies and establish explanation rules. Introducing a step that is not specified and elaborating on its imagined content represents a significant deviation from the prompt and as a consequence must be considered highly irrelevant.

**Criterion: Adherence to instructions and patterns**

**Score 2:** two of three core behavioral rules were only partially or not at all implemented.

The fact check pattern: was only applied in step 2 and 3.1, but not consistently throughout the entire interaction. The reflection pattern: was not applied in step 1, step 2 or step 3. The template pattern: was consistently present throughout the entire interaction. The persona pattern: was consistently present throughout the entire interaction. The meta-language pattern: was evident, as the model explicitly referred to the terminology throughout the entire interaction. The flipped interaction pattern: was observed in step 2, where the model correctly asked the five predefined questions. The cognitive verifier pattern: was applied through the use of sub-questions in step 2.

## 4.4 Comparative Analysis of Model Scores

This section provides a comparative analysis of the performance of four LLMs in assisting with the integration of the SENSE explainability stack. The results shown in Table 5 indicate significant differences in how effectively each model adhered to the prompt and provided structured outputs.

Criteria	Deepseek-r1 (70B)	LLaMA 3.3 (70B)	GPT-4	GPT-4 Turbo
Completeness	1	3	3	5
Relevance	2	3	4	2
Adherence	1	2	2	2

Table 5: Comparative Scores of LLMs Across Evaluation Criteria

**Deepseek-r1 (70B)** had the lowest performance, scoring only one in *completeness*, two in *relevance* and one in *adherence*. It failed to introduce itself correctly, misinterpreted its role and did not make use of predefined terminology definitions. The model did not follow key instructions such as the three behavioral rules and did not adhere to required patterns. Even after user correction, it continued to exhibit inconsistencies.

**LLaMA 3.3 (70B)** performed moderately well, achieving a *completeness* score of three, a *relevance* score of three and an *adherence* score of two. The model introduced itself appropriately and followed most structured steps. Occasionally, it deviated by providing suggested answers instead of asking predefined questions. While it maintained a high degree of relevance, the LLM did not fully adhere to all instructional patterns while showing some inconsistencies.

**GPT-4** achieved strong overall performance, particularly excelling in *relevance* with a score of four. The LLM performed moderately well in *completeness* with a score of three. It introduced itself correctly and maintained a structural approach while following most of the prompt’s guidelines. However, it did not strictly adhere to all predefined instructions, scoring two in *adherence*. The model showed weaknesses in using predefined questions and explicitly asking for types of platforms and sensors.

**GPT-4 Turbo** reached the highest score in *completeness* with a perfect score of five. The model covered all required elements of the prompt, including sub-steps 3.1 and 3.2. However, it scored only two points in *relevance* due to the introduction of a non-existent step 4. This deviation affected the overall achieving score drastically. The model did not apply all behavioral rules consistently, thus scoring only two in *adherence*.

**Completeness:** GPT-4 Turbo provided the most thorough responses by explicitly addressing all steps and sub-steps of the prompt. GPT-4 covered nearly all required aspects of the prompt, with minor omissions in explicitly asking for specific platform and sensor types. LLaMA 3.3 (70B) addressed

most elements but lacked depth in Step 2 and 3. Deepseek-r1 (70B) failed to include major prompt elements, leading to a significantly lower score.

**Relevance:** GPT-4 maintained strict focus and correctly categorized user input, ensuring its responses aligned with the prompt’s requirements. LLaMA 3.3 (70B), while generally relevant, occasionally introduced unnecessary suggestions instead of strictly following the predefined structure. GPT-4 Turbo introduced an entirely fabricated step, negatively affecting the clarity and scope. Deepseek-r1 (70B) exhibited logical relevance but failed to correctly address the system integration task in the initial attempts.

**Adherence to Instructions and Patterns:** None of the models fully adhered to all predefined instructions and patterns. GPT-4 Turbo, GPT-4 and LLaMA 3.3 (70B) achieved a partial level of compliance, but Deepseek-r1 (70B) failed to implement key elements such as fact-checking, reflection and structured output formatting.

The evaluation highlights GPT-4 Turbo as the model with the most complete and structured output achieved, outclassing in *completeness* but facing setbacks in *relevance* due to inclusion of off-topic content. GPT-4 remains the most balanced model in terms of *relevance* and overall structure.

## 5 Discussion and Conclusion

The goal of this thesis was to develop a prompt that supports users - particularly without prior knowledge in semantic technologies - in implementing the SENSE guideline with the help of a large language model (LLM). Through the use of prompt engineering techniques, we aimed to simplify the process of knowledge elicitation in the context of data gathering. Special emphasis was placed on reusable prompt structures and CSV-compatible outputs that can be copied directly into the official SENSE Excel template. The approach is demonstrated in a cyber-physical systems (CPS) context, where explainability and transparency are crucial.

This thesis was guided by the following two research questions:

- *How can existing prompt engineering techniques be synergistically integrated to consistently generate accurate and structured outputs that conform to the SENSE guideline for knowledge graph construction?* [RQ1]
- *To what extent do Deepseek-r1 (70B), LLaMA 3.3 (70B), GPT-4 and GPT-4 Turbo differ in terms of output completeness, relevance, and*



*adherence to instructions and patterns when assisting with knowledge graph construction?* [RQ2]

With regard to *RQ1*, it can be seen that the targeted and synergistic integration of established prompt engineering techniques makes a significant contribution to generating precise and structured output that meets the requirements of the SENSE guideline. By combining various methods - such as the establishment of a uniform technical language (Meta Language Creation), the active involvement of users through targeted questions (Flipped Interaction), the consistent assignment of roles (Persona) and the systematic checking and formatting of input (Cognitive Verifier, Fact Check List, Reflection and Template) - it was possible to create a stable yet flexible interaction framework. This integrative approach not only simplifies the complex process of knowledge acquisition, but also ensures consistent and comprehensible documentation of the system information.

At the same time, the comprehensive use of numerous patterns results in an increased prompt length and adds processing complexity. In certain application scenarios, a reduced, more focused version of the prompt could therefore be advantageous. Overall, however, the thesis underlines that the careful coordination and combination of the techniques used is crucial for achieving high-quality results that meet the high requirements of the guideline in terms of both content and form.

To answer *RQ2*, we tasked the developed prompt on four different LLMs based on three criteria: *completeness*, *relevance* and *adherence to instructions and patterns*. This showed that GPT-4 Turbo achieved the highest completeness (5 out of 5 points) - it covered all the steps specified in the prompt and consistently asked the user for all the required information. However, GPT-4 Turbo occasionally deviated from the prompt by introducing an undesired, non-existent fourth step, which had a negative impact on *relevance* (score: 2). GPT-4, on the other hand, proved to be particularly balanced, providing precise and contextual answers overall (*relevance*: 4) and largely adhered to the required step structure (*Completeness*: 3).

LLaMA 3.3 (70B) and Deepseek-r1 (70B) performed significantly weaker. LLaMA 3.3 (70B) (*Completeness*: 3) often lacked crucial follow-up questions or omitted parts of the step-by-step instructions. Deepseek-r1 (70B) even only achieved a 1 in *completeness*, as it skipped essential parts of the prompt workflow and only insufficiently responded to the input requested by the user. In terms of *relevance*, LLaMA 3.3 (70B) and Deepseek-r1 (70B) were in the medium to low range; in some cases they provided generic or inappropriate responses, which made it difficult to map the SENSE guideline.

In terms of *adherence to instructions and patterns* (e.g. fact-check lists,

reflection and CSV format), none of the models were completely convincing. Nevertheless, GPT-4 Turbo and GPT-4 showed the best approximation here, as they applied at least parts of the rules consistently and generated mostly usable CSV-outputs. LLaMA 3.3 (70B) and Deepseek-r1 (70B) completely ignored important aspects such as fact-check lists or spontaneously switched to other formats.

The analysis confirms that the four models differ significantly in their performance. GPT-4 Turbo excels in complete coverage of the individual steps, while GPT-4 offers the most reliable overall balance of *completeness*, *relevance* and *adherence*. In addition, qualitative differences in the model responses show that even with identical ratings in individual criteria, one model can still perform better than another - despite comparable numerical results.

Although the LLMs tested showed clear differences in performance, it can be stated that the concept of prompt patterns works in principle and can give a complex elicitation process a clearly structured approach. At the same time, general limitations of LLMs become apparent, such as the tendency towards hallucinations and, in some cases, off-topic responses.

With this bachelor thesis, the designed prompt contributes to the user-friendly integration of the SENSE explainability stack in real CPS environments. This is achieved on the one hand through the instructions in clearly defined steps and on the other hand through the forced CSV-output, which allows direct copy-paste into the official SENSE Excel template. This represents an important step towards the automated capture and modeling of domain knowledge, as the need for expert knowledge is partially reduced and a structured, reusable prompt infrastructure is provided.

While the developed prompt and its evaluation provide a solid foundation for supporting users in applying the SENSE guideline, several aspects remain open for further exploration. The following section outlines potential directions for future work, including the expansion of evaluation perspectives, refinement of scoring mechanisms, and the influence of model configuration parameters such as temperature.

## 5.1 Future Work

This thesis opens several opportunities for future research in the area of LLM assistance for knowledge elicitation within semantic frameworks such as SENSE.

### 5.1.1 Expanded Evaluation Approaches

While this work used predefined criteria focusing on *completeness*, *relevance* and *adherence to instructions and patterns*, future evaluations could incorporate additional dimensions to gain deeper insights into model performance. Examples could include:

- **Faithfulness:** Checks whether the model outputs demonstrably match the user data and the system description. This ensures that the generated content is reliable and true to the data.
- **Cognitive effort:** Assesses whether the support provided by the model reduces the user's mental load. Measuring cognitive effort can reveal whether the prompt noticeably simplifies the data collection process and improves usability.
- **Robustness with inconsistent inputs:** Evaluates how reliably the model responds to contradictory, incomplete or ambiguous user information - typical challenges of real data collection. The focus is on whether the model asks questions, recognizes ambiguities and still delivers structured, usable results.

Additionally, involving user studies could provide valuable feedback on usability, trust and perceived intelligence of the LLM.

### 5.1.2 Refinement of Scoring System

The current scoring scale was applied uniformly across three criteria points. Further refinements could increase precision and reduce subjectivity. Further research may explore:

- **Weighted scoring:** Assign greater weight to critical behaviors like adherence to behavioral rules.
- **Rubric expansion:** Break down each criterion into micro dimensions (e.g., for criterion "adherence": fact-checking, reflection, formatting) and scoring each individually.
- **Time based metrics:** Track how quickly or efficiently users reach satisfactory outputs with the best performing model.

In addition to possible extensions, another limitation of the current rating system became clear. Although models can receive the same score in one criterion, they sometimes differ significantly in the actual quality of their performance. An example: Two models each receive a score of three in the *adherence* criterion. While one model applies most of the given patterns consistently, but has difficulty maintaining the role persona throughout, the other model implements several interaction patterns only superficially and inconsistently. Such qualitative differences can lead to weaker performing models being overestimated in the overall impression. A possible improvement would be the introduction of dynamic weightings based on specific behaviors. This could contribute to a more differentiated and accurate assessment of model behavior.

### 5.1.3 Impact of Model Temperature on Prompt Results

*Peeperkorn et al.* [16] investigate whether the temperature parameter in LLMs truly functions as a creativity parameter, which is a common claim in AI. The findings reveal a weak correlation between temperature and coherence and thus, higher temperature slightly increases novelty but at cost of coherence. The authors argue that temperature alone is not a reliable proxy for creativity.

*Renze et al.* [18] explore how temperature affects problem solving accuracy of LLMs on multiple choice tasks across various domains. The authors find that changing temperature from 0.0 to 1.0 does not significantly impact accuracy. The paper states that temperature adjustments within this range do not improve LLM performance on problem-solving tasks. For tasks where accuracy and responsibility are key, the authors recommend using a temperature of 0.0.

Based on the results of both studies, future research could investigate the influence of the temperature parameter in tasks that go beyond creative text generation or multiple-choice questions. In particular, for multi-step prompts such as the SENSE integration task, variation in temperature could provide insight into whether a lower temperature value improves compliance with instructions. A systematic experiment could compare outputs at different temperature values - for example in sub-steps such as the explanation of terms or the processing of user questions. In this way, it would be possible to empirically test whether and to what extent the temperature setting influences the consistency of answers in application scenarios for system integration.

## References

- [1] S. Amershi, D. Weld, M. Vorvoreanu, A. Fournery, B. Nushi, P. Collisson, J. Suh, S. Iqbal, P. N. Bennett, K. Inkpen, J. Teevan, R. Kikin-Gil, and E. Horvitz. Guidelines for human-ai interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, Paper 3, 1–13. <https://doi.org/10.1145/3290605.3300233>, 2019.
- [2] P. R. Aryan, F. J. Ekaputra, M. Sabou, D. Hauer, R. Mosshammer, A. Einfalt, T. Miksa, and A. Rauber. Explainable cyber-physical energy systems based on knowledge graph. In *Proceedings of the 9th Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MCPES'21)* (pp. 1-6). ACM. <https://doi.org/10.1145/3470481.3472704>, 2021.
- [3] M. Broy, M. V. Cengarle, and E. Geisberger. Cyber-physical systems: Imminent challenges. In: *Calinescu, R., Garlan, D. (eds) Large-Scale Complex IT Systems. Development, Operation and Management. Monterey Workshop 2012. Lecture Notes in Computer Science, vol 7539*. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-34059-8\\_1](https://doi.org/10.1007/978-3-642-34059-8_1), 2012.
- [4] S. Chari, D. M. Gruen, O. Seneviratne, and D. L. McGuinness. Directions for explainable knowledge-enabled systems. In *Knowledge Graphs for Explainable Artificial Intelligence: Foundations, Applications and Challenges (Studies on the Semantic Web, Vol. 47, pp. 245–261)*. IOS Press. <https://doi.org/10.3233/SSW200022>, 2020.
- [5] A. Drechsler and A. Hevner. A four-cycle model of is design science research: capturing the dynamic nature of is artifact design. In: *Parsons, J., Tuunanen, T., Venable, J. R., Helfert, M., Donnellan, B., Kenneally, J. (eds.) Breakthroughs and Emerging Insights from Ongoing Design Science Projects: Research-in-progress papers and poster presentations from the 11th International Conference on Design Science Research in Information Systems and Technology (DESRIST) 2016. St. John, Canada, 23-25 May. pp. 1-8*, 2016.
- [6] T. Gavrilova and T. Andreeva. Knowledge elicitation techniques in a knowledge management context. *Journal of Knowledge Management*, 16(4), 523 - 537. <https://doi.org/10.1108/13673271211246112>, 2012.

- [7] L. Giray. Prompt engineering with chatgpt: A guide for academic writers. *annals of biomedical engineering*. 51(2629-2633). <https://doi.org/10.1007/s10439-023-03272-4>, 2023.
- [8] Alan R. Hevner. A three cycle view of design science research. *Scandinavian Journal of Information Systems: Vol. 19 : Iss. 2 , Article 4*. Available at: <http://aisel.aisnet.org/sjis/vol19/iss2/4>, 2007.
- [9] A. Hogan, E. Blomqvist, M. Cochez, C. D’Amato, G. de Melo, C. Gutierrez, S. Kirrane, J. E. Labra Gayo, R. Navigli, S. Neumeier, A.-C. Ngonga Ngomo, A. Polleres, S. M. Rashid, A. Rula, and L. Schmelzeisen. Knowledge graphs. *ACM Computing Surveys*, 54(4), Article 71. <https://doi.org/10.1145/3447772>, 2021.
- [10] S. S. Jha, S. Mayer, and K. García. Poster: Towards explaining the effects of contextual influences on cyber-physical systems. *11th International Conference on the Internet of Things (IoT '21), November 8–12, 2021, St.Gallen, Switzerland*. ACM, New York, NY, USA. <https://doi.org/10.1145/3494322.3494359>, 2021.
- [11] D. Kerrigan, J. Hullman, and E. Bertini. A survey of domain knowledge elicitation in applied machine learning. *Multimodal Technologies and Interaction*, 5(12), 73. <https://doi.org/10.3390/mti5120073>, 2021.
- [12] V. K. Kommineni, B. König-Ries, and S. Samuel. From human experts to machines: An llm supported approach to ontology and knowledge graph construction. *arXiv*. <https://arxiv.org/abs/2403.08345>, 2024.
- [13] J. Y. Mao, K. Vredenburg, P. W. Smith, and T. Carey. User-centered design methods in practice: a survey of the state of the art. In *Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research (p. 12)*, 2001.
- [14] G. Marvin, N. Hellen, D. Jjingo, and J. Nakatumba-Nabende. Prompt engineering in large language models. In *I. J. Jacob et al. (Eds.), Data intelligence and cognitive informatics (pp. 387-412)*. Springer Nature. <https://doi.org/10.1007/978-981-99-7962-2>, 2024.
- [15] A. O’Hagan. Expert knowledge elicitation: Subjective but scientific. *The American Statistician*, 73:sup1, 69-81, DOI: 10.1080/00031305.2018.1518265, 2019.

- [16] M. Peeperkorn, T. Kouwenhoven, D. Brown, and A. Jordanous. Is temperature the creativity parameter of large language models? *arXiv preprint arXiv:2405.00492*. <https://arxiv.org/abs/2405.00492>, 2024.
- [17] L. Petnga and M. Austin. An ontological framework for knowledge modeling and decision support in cyber-physical systems. *Advanced Engineering Informatics, Volume 30, Issue 1, 2016, Pages 77-94, ISSN 1474-0346*, <https://doi.org/10.1016/j.aei.2015.12.003>, 2016.
- [18] M. Renze and E. Guven. The effect of sampling temperature on problem solving in large language models. *In Findings of the Association for Computational Linguistics: EMNLP 2024 (pp. 7346-7356)*., 2024.
- [19] F. Maurer T. Silva da Silva, A. Martin and M. Silveira. User-centered design and agile methods: A systematic review. *2011 Agile Conference, Salt Lake City, UT, USA, 2011, pp. 77-86, doi: 10.1109/AG-ILE.2011.24*, 2011.
- [20] J. Wang, W. Ma, P. Sun, M. Zhang, and J. Nie. Understanding user experience in large language model interactions. <https://doi.org/10.48550/arXiv.2401.08329>, 2024.
- [21] 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. <https://arxiv.org/abs/2302.11382>, 2023.

## A Appendix

This section grants access to all LLM interactions via a shared GitHub repository. A direct chat link is provided for the GPT-4 Turbo interaction.

Github:

<https://github.com/JonHemCPS/LLM-Screenshots/tree/main>

GPT-4 Turbo Chat:

<https://chatgpt.com/share/67e56450-565c-8002-964a-32a4f600de18>

The full prompt is included in this appendix for reference.



**You will guide me through integrating the “SENSE Explainability Stack” into my system.** Always maintain the persona of a dedicated assistant focused on accurately capturing the system's structure and causal/state relationships, ensuring that I am supported when I enter the data that describe the system.

I am following the “SENSE User Guideline” to extract information from my existing system.

To help you better understand the context, here are the key points:

1. The **SENSE User Guideline** is designed to help users integrate the *SENSE Explainability Stack* into an existing system.
2. A **common conceptualization** is crucial to the successful integration of the SENSE Explainability Stack.
3. **Definitions of terms** used within the guideline are essential for maintaining a shared understanding. These definitions **must be followed precisely** at all times.
4. When I ask questions, **adhere** to these definitions and rules to ensure consistency and correctness.

Let's define a concise “meta-language” for clarity. Whenever the following terms are used, they carry the meanings below:

## **Platform**

A platform is any *device, facility, or logical group* that can host sensors measuring data. Devices, facilities, and logical groupings are all considered **platforms** in the SENSE stack.

## **PlatformType**

Each platform belongs to a specific PlatformType, such as a **household**, **battery**, or **EV charger**.

## **Sensor**

A sensor measures some property in the system and is **always hosted by a platform**. Each sensor is physically or logically located on exactly one platform.

## **SensorType**

A category of sensors that measure specific properties, e.g., **ActivePowerSensor** (ActivePower), **StateOfChargeSensor** (StateOfCharge).

## **hostedBy**

A **hierarchical relationship** indicating that a sub-platform (or a sensor) is hosted by a higher-level platform. In the SENSE stack, the first (top-level) platform is typically considered the “host,” and any nested platforms or sensors are said to be **hostedBy** it.

I establish three behavioral rules that you must follow at all times:

### **Rule 1:**

After each major section (Step 2 and Step 3), please provide a **fact check list** of critical statements or assumptions I have made. Label them as:

**Fact to Check:** "X."

**Potential Source or Reason:** "Why we need to verify it."

**Possible Consequence:** "What might happen if it's incorrect."

For example:

**Fact to Check:** "The EV Charger can always cause a peak demand."

**Potential Source:** I observed large energy spikes in the past.

**Possible Consequence:** If untrue, we might overestimate the battery's impact.

This helps to ensure accuracy before we lock in any detail.

### **Rule 2:**

After you provide any answer (except: fact check list), add a short **Reflection** section:

Explain **why** you arrived at that specific conclusion or structure.

Clarify any assumptions or inferences you made.

Example:

#### **Reflection:**

"I noticed you mentioned a battery but didn't confirm its type. I assumed it was a standard Lithium-Ion battery because of the mention of 'StateOfChargeSensor.' If that assumption is incorrect, let me know, and I'll adjust accordingly."

This helps me see your reasoning process and correct any misunderstandings.

### **Rule 3:**

Whenever I provide data, please format it in a tab-delimited table. Use this exact syntax (CSV-format):

Header1;Header2

Value1;Value2

...

The first row is for column headers, and each subsequent row must be aligned under the correct header, leaving empty fields where appropriate. Make sure I can simply copy and paste your entire output to view it as a neatly arranged table.

In the following, I will show you the steps that need to be followed in order to integrate the “SENSE Explainability Stack”. Along the way, there will be text passages, indicating an example use case (toy example). The toy example passages will follow through each step, each time adding a new layer of information to build a schema.

## Step 1

Make me aware of your role and what you want to help me with.

Give me a brief overview of the terminology used in the guideline and its definitions. Additionally, briefly make me aware of the three rules you follow.

Ask me if I am ready for step 2 before you begin with the next step.

## Step 2

First, I need to define the goals of using the “SENSE Explainability stack” in my system.

I want you to **ask me questions** so we can collect all relevant information step by step. Specifically:

Which questions should be answered by the system?  
What are important anomalies you want to detect and explain?  
What is the purpose of doing this?  
What is the end result we expect?  
What are the limitations and scope?

When you see my answers, please:

1. Break them down into smaller sub-questions or points **if needed** for clarity.
2. Confirm each sub-point to ensure there are no misunderstandings.
3. Synthesize the final structured information (platforms, sensors, connections, etc.) before moving on

The answers to these questions should always be the guideline for any decision in the next steps.

If needed, you can give me the following toy example for clarity:

We consider a Household, which contains an Electric Vehicle(EV) Charger and a Battery. If the EV Charger is fast-charging an EV that is plugged in, it can cause a peak demand at the household level. However, this peak only happens when it is enabled by an empty Battery.

Ask me if I am ready for step 3 before you begin with the next step.

## Step 3

The goal of this step is to have a representation of my system to be analysed. Therefore, a list of devices, facilities, logical groupings in my system as well as sensors collecting data has to be collected.

Ask me to list **all platforms** in my system.

Ask me for the **hostedBy connections** between these platforms.

Ask me to list **all sensors** in my system.

Ask me to list **all connections** between **sensors** and **platforms** (to map each sensor to its relevant platform)

### Step 3.1

Ask me to define the **types** of the platforms, sensors and the *observable properties* the sensors measure.

The toy example from Step 1 mentions a household (PlatformType), a battery (PlatformType) and an EV Charger (PlatformType). All three are PlatformTypes.

Table in CSV-format:

```
PlatformType;  
household;  
battery;  
evcharger;
```

The toy example from Step 1 is enhanced with an ActivePowerSensor (SensorType) and a StateOfChargeSensor (SensorType). Both are SensorTypes.

Table in CSV-format:

```
SensorType;  
ActivePowerSensor;  
StateOfChargeSensor;
```

The toy example from Step 1 defines observable properties:

ActivePowerSensor has the observable property ActivePower  
StateOfChargeSensor has the observable property StateOfCharge

### Step 3.2

Ask me for the **actual instances** of each platform and sensor.

Ask how they connect hierarchically (using hostedBy).

Ask me to confirm which observable property each sensor measures.

To continue the toy example from Step 1 and Step 2.1:

There is one household platform (household1 of type household), one battery (battery1 of type battery) and one EV Charger (evcharger1 of type evcharger). evcharger1 and battery1 are both hosted by household1 as they are logically part of the household.

Connections between Platforms are defined by the "hosts" relation as well: (household1 hosts evcharger1) and (household1 hosts battery1)

Table in CSV-format:

```
Platform;PlatformType;hostedBy_Platform
household1;household;household1
battery1;battery;household1
evcharger1;evcharger;household1
```

There is one ActivePowerSensor at the household level, measuring the total power used in the household – e.g. a smart meter (AP\_household1\_sensor hosted by household1, observing ActivePower). There is one ActivePowerSensor at the evcharger (AP\_evcharger1\_sensor hosted by evcharger1, observing ActivePower). There is one ActivePowerSensor at the battery (AP\_battery1\_sensor hosted by battery1, observing ActivePower). There is one StateOfChargeSensor at the battery (SOC\_battery1\_sensor hosted by battery1, observing StateOfCharge).

Table in CSV-format:

```
Sensor;SensorType;hostedBy_Platform;observes_ObservableProperty;TimeseriesId
AP_household1_sensor;ActivePowerSensor;household1;ActivePower;
AP_evcharger1_sensor;ActivePowerSensor;evcharger1;ActivePower;
AP_battery1_sensor;ActivePowerSensor;battery1;ActivePower;
SOC_battery1_sensor;StateOfChargeSensor;battery1;StateOfCharge;
```

## List of aids for seminar thesis

**Title of thesis:**

**LLMs for User Assistance in Gathering Domain Knowledge**

**Author:**

**Last name, first name, student ID number:**

**Hemedinger, Jonas Christian, 11933393**

Aids/tools used	Type(s) of use	Affected areas/chapters	Documentation
ChatGPT	ChatGPT was used to stylistically improve individual passages, limited strictly to linguistic and grammatical editing.	Applied throughout the entire thesis (all chapters affected).	Distributed across multiple chat sessions over the course of the thesis work.
ChatGPT	ChatGPT was used to generate and adapt shell commands to support communication with the virtual server provided by the university.	Not directly reflected in the chapters, but used during the implementation setup phase.	<a href="https://chatgpt.com/share/67f7c496-c074-8012-b718-a92c799193f4">https://chatgpt.com/share/67f7c496-c074-8012-b718-a92c799193f4</a>
ChatGPT	ChatGPT was used to look up and adapt	Not explicitly reflected in the chapters, but applied	<a href="https://chatgpt.com/share/67f7bf09-822c-8012-8385-e6d1bb42bc50">https://chatgpt.com/share/67f7bf09-822c-8012-8385-e6d1bb42bc50</a>

	<b>individual LaTeX commands for formatting purposes (e.g., tables, citations, figure placement).</b>	<b>throughout the document for layout and formatting adjustments.</b>	
--	---	---	--

I hereby declare that I have listed all the aids I have used in the list above. If no aids have been used, it is also indicated in the list (to be listed under “Aids/tools used: none”).

05/05/2025

Date

J. Hem

Signature(s)