Ontology Engineering with Large Language Models

Unveiling the potential of human-LLM collaboration in the ontology extension process

García-Fernández, Julia; Verhoosel, Jack; Ubacht, Jolien; Bakker, Roos Marieke

Publication date
2025

Document Version
Final published version

Published in
CEUR Workshop Proceedings

Ontology Engineering with Large Language Models: Unveiling the potential of human-LLM collaboration in the ontology extension process

Julia García-Fernández[1,*,†], Jack Verhoosel[1,3,†], Jolien Ubacht[2,†] and Roos Marieke Bakker[1,4,†]

[1]Data Science Department, The Netherlands Organization for Applied Scientific Research (TNO), Anna van Buerenplein 1, 2595 DA The Hague, The Netherlands

[3]Institute of Data Science, Maastricht University, Paul-Henri Spaaklaan 1, 6229 GT Maastricht, The Netherlands

[2]Faculty of Technology, Policy and Management, Delft University of Technology, Jaffalaan 5, 2628 BX Delft, The Netherlands

[4]Leiden University Centre for Linguistics, Leiden University, Cleveringaplaats 1, 2311 BD Leiden, The Netherlands

Abstract

In this paper, we present a domain-independent ontology extension workflow supported by LLMs. Ontology Engineering (OE) is a complex field that requires combining technical skills with domain expertise across multiple disciplines. Despite numerous attempts at automation, most of the processes are still manual. Different ontology engineering methodologies coexist, but none is a standard. These challenges, together with the lack of highly skilled workers in the sector, increase the entry barriers to the field. In parallel, Large Language Models (LLMs) are becoming prominent in ontology development due to their natural language processing and coding capabilities and their reportedly emergent abilities. In this paper, we focus on human-LLM collaboration for ontology extension. Following a Design Science Research approach, we interviewed 11 experts and modeled the current process of ontology extension to disclose its main issues. We analyzed the concerns and opportunities perceived by ontology engineers for using LLMs. Based on our insights and previous work, we designed a process framework for ontology extension that combines human expertise with LLMs capabilities, providing customizable prompt templates, OE tools, and guidelines. We tested our methodology with an existing greenhouse ontology using GPT-4o. Finally, we qualitatively evaluated the results against a manually crafted extension we use as our gold standard. The results show that the proposed approach holds the potential to (1) get inspiration for adding new entities, (2) deal with complex syntax definitions and repetitive tasks, and (3) verify whether the extended ontology conforms to the requirements and competency questions.

Keywords

Ontology Engineering, Large Language Models, Human-in-the-loop, Ontology Learning

1. Introduction

Ontologies are increasingly developed and used in many domains and sectors to unambiguously define the semantics of concepts and their relations [1]. They are widely used in information systems where data must be automatically interpreted, not only by humans but also by machines [2]. In the traditional way of Ontology Engineering (OE), an ontology engineer works together with a domain expert to determine the main concepts of a domain and how they fit together. This is a time-consuming process in which often the definition of a concept or term is iteratively fine-tuned manually to capture its exact meaning [3].

Since many domains already have an ontology defined, the next challenge lies in managing its changes and extensions. In addition to difficulties in defining new concepts and relations, there are also questions about where and how these new elements should be integrated within the existing ontology. This makes the engineering process even more complex, as definitions of new concepts now have to

LLM-TEXT2KG 2025: 4th International Workshop on LLM-Integrated Knowledge Graph Generation from Text (Text2KG), June 1 - June 5, 2025, co-located with the Extended Semantic Web Conference (ESWC 2025), Portorož, Slovenia.

*Corresponding author.

julia.garciafernandez@tno.nl (J. García-Fernández); jack.verhoosel@tno.nl (J. Verhoosel); j.ubacht@tudelft.nl (J. Ubacht); roos.bakker@tno.nl (R. M. Bakker)

0000-0002-0121-636X (J. Verhoosel); 0000-0002-1269-7189 (J. Ubacht); 0000-0002-1760-2740 (R. M. Bakker)

adhere to already existing definitions in the ontology. An unfortunate side trend is that there is a lack of highly skilled ontology engineers, so tackling this laborious job becomes even more difficult [4].

To deal with this challenge, one of the solution directions is to make use of Large Language Models (LLMs) that have arisen in the last few years as a potentially helpful tool for complex language tasks [5, 6, 7, 8]. Therefore, this paper focuses on the main question of how LLMs can be of use and in which OE tasks, helping the ontology engineer and/or the domain expert in extending their ontology in a human-LLM collaborative manner.

To answer this question, we applied the Design Science Research (DSR) approach to (1) investigate the main challenges in ontology extension, (2) gather requirements for a human-LLM collaborative ontology extension process framework, (3) determine which tasks can best be supported by an LLM, (4) build a prototype that implements the application of an LLM in the ontology extension process framework and (5) evaluate its usability with a specific use case for an existing ontology in the greenhouse sector.

The main contributions of our work are (1) An analysis of the complexity of the process of ontology extension, the main concerns and identified opportunities for the use of LLMs for this process as perceived by the ontology engineers; (2) a set of 22 high-level requirements for the design of an LLM-assisted process framework for ontology extension; (3) a domain-independent workflow that integrates LLMs in the process of ontology extension combining the human expertise with the LLM's capabilities; (4) a qualitative evaluation of LLM performance across various ontology extension tasks, highlighting those with the best outcomes and the greatest potential for practical implementation. Supplementary material, including the data gathered through the interviews, the process framework, the prompt templates, and all the inputs and outputs from the demonstration and evaluation with GPT-4o can be found on GitLab[1].

In Section 2, we outline related work on OE in combination with the field of Natural Language Processing. Next, in Section 3 we describe our methodology. In Section 4, we introduce our results and the domain-independent ontology extension workflow. In Section 5, the framework is evaluated. Finally, we summarize our work and explore future directions in Section 6.

2. Related Work

Ontologies are formal models that describe concepts and relations of a domain, and are traditionally created and maintained manually by domain experts and ontology developers [9]. This process is time-intensive, error-prone, and costly to maintain [7]. To address these challenges, multiple efforts have been made to automate this process, or parts of them, using a range of techniques. Early work identified a range of tasks ranging from term extraction to learning axioms, relying heavily on rule-based and lexico-syntactic methods [10]. More recently, statistical methods were developed such as co-occurrences and hierarchical clustering, pushing the performance of automatic ontology development [11, 12].

In recent years, advances in Natural Language Processing (NLP) have accelerated the possibilities of automating parts of the OE process, with techniques combining linguistic and statistical methods [13]. With the introduction of LLMs, such as GPT [14], new opportunities for automating specific ontology learning tasks have risen, including term typing, taxonomy discovery, and non-taxonomic relation extraction [7, 15].

Since the introduction of LLMs, there have been many attempts to apply them to the OE process [16, 17, 18]. Different approaches of applying LLMs can be distinguished: (1) generate ontologies or Knowledge Graphs (KGs) end-to-end with unstructured [19] or semi-structured data [17, 20], or (2) generate parts of ontologies or KGs in a multi-step approach.

Examples of the first are recent works that input raw text, prompt an LLM for extraction, and evaluate the resulting model against a ground truth [21, 22]. Bakker et al. [22] concluded that the results are still far from the manual ground truth, and an approach where human domain expertise is combined with the LLM might lead to better results. An example of this is the approach proposed by Saeedizade and Blomqvist [17], who tested different techniques from zero-shot prompting to decomposed prompting

[1] https://gitlab.com/eswc2025/ontology-extension-with-llms

and chain-of-thoughts to instruct an LLM how to generate an OWL ontology. The authors conclude that the ontologies generated by the combination of GPT-4 with advanced prompting techniques are comparable to ontologies manually crafted by beginner ontology engineers.

Examples of the second approach are the generation of Competency Questions [23, 24], and different downstream tasks such as Relation Extraction [25, 7], Information Extraction [26] or SPARQL query generation and KG population [27]. Although LLMs continue to exhibit shortcomings in these tasks, tackling a single OE activity at a time provides more control to the human over the OE process compared to the approaches previously mentioned.

Additionally, pipelines and frameworks that combine prompting and other techniques are introduced. For instance, [16] developed a domain-agnostic prompting pipeline based on the NeOn methodology [28], called NeOn-GPT. Using GPT-3.5, they generated a wine ontology and compared it to a gold standard, evaluating structural metrics and modeling decisions. While their results highlight the potential of LLMs to support ontology development, they emphasize that human expertise remains essential for achieving the depth and precision of traditional OE. Similarly, [18] introduced OntoChat, a conversational framework for tasks such as requirements elicitation, competency question analysis, and testing, based on input from ontology engineers and domain experts. Evaluated with a musical ontology, OntoChat was well-received for reducing manual effort in these three tasks despite acknowledged limitations, showcasing the promise of LLMs to streamline challenging aspects of OE.

A common aspect of recent studies is that LLMs do not create ontologies that are of sufficient quality [16, 22]. Solutions such as OntoChat [18] show the potential of a hybrid approach, where LLMs and domain experts work together on creating an ontology. The question remains open as to how to integrate LLMs in the OE process in practice, so that LLMs become a valuable tool in the OE toolkit.

3. Methodology

For the development of the process framework that can reduce the complexity in the ontology extension process by using LLMs, we followed the Design Science Research approach (DSR). The DSR approach has gained common ground in the information systems domain through the seminal works of [29], [30], and, more recently, [31], who provide a practical, phased way of working on designing artifacts. The DSR encompasses the phases described below.

3.1. First phase: Explicate problem

In our research, the aim of the first DSR phase of problem explication was to analyze the current manual process that ontology engineers follow to extend existing ontologies and to uncover the issues they experience during their activities. We followed the Human Research Ethics Research Design Plan of a known research institution, which is in line with European guidance on research ethics. This included a Data Management Plan, a Risk Assessment and Mitigation Plan, and an Informed Consent Procedure. The design plan was submitted and approved before starting the interview process.

We conducted semi-structured interviews with 11 professionals in applied research in OE and LLMs to map their daily practices and to analyze the root causes of difficulties they encounter during the ontology extension process. From the 11 interviewees, 10 are ontology engineers and 1 is mainly focused on LLMs and NLP. From the ontology engineers, some have a background in Artificial Intelligence and NLP, others have a background in formal logic, and some are more focused on operational ontologies and consultancy in the semantics and standardization sector. As reported by the interviewees, they do not normally use OE methodologies in practice, although some of them used SABiO [32] before. They have a customized OE approach and set of best practices, and 6 out the 11 often use Competency Questions.

Throughout these interviews, we explored how the integration of LLMs in the ontology extension workflow can be established. Their input led to the decision to design a human-LLM collaboration framework, emphasizing that LLMs can provide added value for certain OE tasks but not fully automate

them. This requires a critical assessment of the LLMs along the way and an approach that is informative but not normative.

3.2. Second phase: Define requirements

The second phase in the DSR approach was to develop a set of functional and non-functional requirements for the process framework design. As the academic literature was still too limited to elicit requirements for our framework, we used the input from the interviews in the previous phase to elicit functional and non-functional requirements and to understand the current process of ontology extension.

We modeled the current ontology extension process (i.e., the manual extension process without the use of LLMs) based on the responses to the interviewees in the previous phase. They were asked several questions about their current OE process, including they steps they execute, the tools and methods they use, and the stakeholders they normally collaborate with. The result of this modeling process was a flowchart with OE phases, including the actual activities that ontology engineers conduct to develop an extension, the OE tools they use, and several stakeholders involved in some of the activities of the process.

After conducting the interviews, eliciting an initial list of requirements, and modeling the current ontology extension process, we organized a focus group session. All the interviewees that previously participated were invited to the focus group session (but not all of them were present). This session was aimed to validate both the previously elicited requirements and the ontology extension workflow generated. To validate the requirements, we conducted a live survey where we asked "What are the requirements for a human-LLM collaboration framework for ontology extension?". The participants' responses where shown in the session and we asked them to vote for their preferred answers. We saved the answers and votes in a table and used this table to validate the requirements previously elicited from the interviews transcripts and to elicit new requirements. The final list of requirements is provided in Table 2 (Appendix A.1).

3.3. Third phase: Design and develop artifact

In the third phase of the DSR approach, we designed a prototype of the process framework, taking the requirements into account. As we wanted the framework to support ontology engineers by not only guiding the ontology extension process steps and activities but also integrating the LLMs for each activity, we mapped the NLP capabilities of LLMs to the downstream tasks in the current ontology extension process. We also analyzed recent literature on the application of LLMs in OE (which, during our research, has been increasingly growing). Our analysis provided an overview of the current use of LLMs for OE tasks, existing LLM-based methods and tools, configurations, and prompt engineering techniques. We arrived at a prototype of the comprehensive process framework for ontology extension by integrating these findings into a process framework design.

3.4. Fourth and fifth phases: Demonstrate and evaluate artifact

In our last research phase in the DSR approach, we chose the use case Semantic Explanation and Navigation System (SENS)[2] to demonstrate and evaluate our process framework. Within the project SENS, the Common Greenhouse Ontology (CGO)[3] [33, 34] was extended with concepts and relations about autonomous systems working and navigating in the greenhouse. The SENS extension to the CGO was previously derived manually without using LLMs or LLM-based tools. Therefore, we used the manually generated extension as the gold standard for comparison with the extension produced by using our prototype.

[2] https://appl-ai-tno.nl/projects/sens
[3] https://gitlab.com/ddings/common-greenhouse-ontology

Recent work on the application of LLMs for OE tasks evaluates the performance of LLMs with quantitative metrics such as precision and recall against a ground truth dataset [7, 26, 35, 23, 36, 22]. Although this is a scalable and perhaps more objective way to assess the performance, it is important to consider that when extending an ontology in a real setting, there is no ground truth or gold standard available. Even if a gold standard exists (e.g., when evaluating whether the LLMs can generate the same extension or the same set of manually formulated CQs), there is no single correct way to model an ontology [37] or generate a CQ. Consequently, it is challenging to define a specific set of criteria that can be used to assess the quality of ontologies [38]. Furthermore, since ontologies are updated regularly, evaluating the quality of the introduced changes is crucial [39]. This is especially relevant to the ontology extension case, where the extension's quality should be measured relative to the quality of the existing ontology to be extended. Even if some authors make a good attempt at generating their own metrics to assess the quality of the ontology generated by the LLMs, these metrics are simple and fail to measure the quality of an ontology. Examples of these are counting the number of classes or axioms [16], or using binary indicators like the presence of an "EquivalentClass" restriction [17]. For these reasons, in this work, we decided to demonstrate and evaluate the process framework design with a focus on the LLM-assisted tasks by applying it to a real use case and qualitatively comparing a manually generated extension to an ontology (our gold standard), with the one generated using the framework.

4. Results

In this section, we first present the results of the interviews, covering different topics regarding the complexity of the ontology extension process and the use of LLMs, in Section 4.1. Next, in Section 4.2, we show the prototype design produced based on the list of requirements elicited from the interviews. The full results can be found in GitLab[4], including the current ontology extension process modeled in a flowchart diagram and visualization of the themes analyzed in the interviews.

4.1. Interviews results

Throughout the interviews, we explored several themes about OE and LLMs. First, we asked all interviewees about the processes or methods they follow when extending an existing ontology, the tools they most often use, and the stakeholders normally involved. Because there is no standard process for OE, nor for extending an existing ontology, we used this information to map as precisely as possible the current ontology extension process. This is described below in Section 4.1.1. In addition, we asked the interviewees about the problems they experience when extending an ontology, the opportunities and the concerns they perceive about the application of LLMs to OE. The results are explained below in Section 4.1.2.

4.1.1. The current ontology extension process

There are multiple methodologies for developing ontologies, but there is no consensus on which one should be the standard. In fact, OE methodologies have the highest impact on the relevance and (re)use of ontologies when these are adapted to the needs of the ontology engineers and the requirements of the project [40].

One of the outcomes of the interviews described in Section 3 is that the process of ontology extension is neither static nor linear. It heavily varies depending on different factors such as time and budget constraints for the project, the type of ontology (i.e., reference or operational ontology), the availability of standards or structured documentation for the domain or specific use case ("Bottom-up approach") or the need to extract the knowledge from the domain experts ("Top-down approach"), and even personal preferences such as the choice of using Compentency Questions (CQs).

[4] https://gitlab.com/eswc2025/ontology-extension-with-llms/-/tree/main/Interviews

Table 1
Summary of problems in OE, and concerns and opportunities for LLMs in OE as identified by the interviewees.

Category	Aspect	Summary
Problems	Existing ontologies are too big/complex	Ontologies grow in size and complexity over time, making governance and management challenging.
	Long discussions with stakeholders	Reaching agreement often involves lengthy debates, especially when domain experts lack technical knowledge.
	High expertise required	Extending ontologies demands high abstraction and familiarity with complex or unfamiliar domains.
	Manual maintenance	OE processes are largely manual; existing tools are often inadequate for efficient workflows.
Concerns	Hallucinations	LLMs sometimes generate plausible but incorrect responses, which can be hard to detect in formal OE.
	Environmental and ethical concerns	LLMs have high energy usage; proprietary models pose risks to data privacy and intellectual property.
	Loss of enriching human process	OE fosters collaboration and shared understanding, which could diminish if fully replaced by LLMs.
Opportunities	Creativity and inspiration	LLMs can generate out-of-the-box ideas, especially during initial phases in OE.
	Entity extraction	LLMs can extract concepts and relationships from unstructured text, aiding information extraction.
	Suggestions for best practices and syntax checking	LLMs could provide real-time advice on best practices and validate syntax during manual OE tasks.
	SPARQL query generation	LLMs can generate SPARQL queries from natural language, bridging gaps in human-machine communication.
	Small and simple OE tasks	LLMs can assist with repetitive or straightforward tasks, reducing manual effort.

To illustrate the process of ontology extension, we modeled the different phases, tasks, stakeholders, tools, and decisions to be made within a flowchart diagram. The phases, inspired by the methodology and set of best practices followed by the interviewees and by well-known methodologies such as SABiO [32] and HCOME [41], are Preparation; Conceptualization; Implementation; Verification; Exploitation; and Validation. Stakeholders include the ontology engineer, the domain experts, and the knowledge worker (here defined as the one responsible for importing and integrating the extended ontology within the information system that makes use of it). According to the interviewees, the most used tools are Protégé and TopBraid, as well as generic code editors and diagramming tools for ad-hoc visualizations of the ontology.

4.1.2. Problems in Ontology Engineering, and challenges and opportunities for using LLMs

The main challenges and opportunities identified in OE can be divided in three categories: OE problems, concerns about LLMs, and opportunities for LLMs in OE. We provided an overview of the aspects discussed within each category in Table 1 below.

An overview of the causes that make the ontology extension process complex is shown in Figure 1. Key problems include the increasing size and complexity of existing ontologies, which complicates governance and system integration. Additionally, lengthy discussions with stakeholders often arise due to difficulties in translating domain knowledge into reusable and standardized ontology elements, particularly when domain experts lack technical expertise. High levels of abstraction and familiarity with complex or unfamiliar domains are required, making the process demanding for engineers. Moreover, OE tasks remain largely manual, with limited availability of robust tools to streamline workflows.

LLMs can provide opportunities for reducing the complexity in the ontology extension process, but it also raised concerns with the interviewees. They include technical, environmental, and ethical dimensions. A significant technical challenge that the interviewees identified is the issue of hallucinations, where LLMs generate responses that seem correct but include inaccuracies. This is particularly

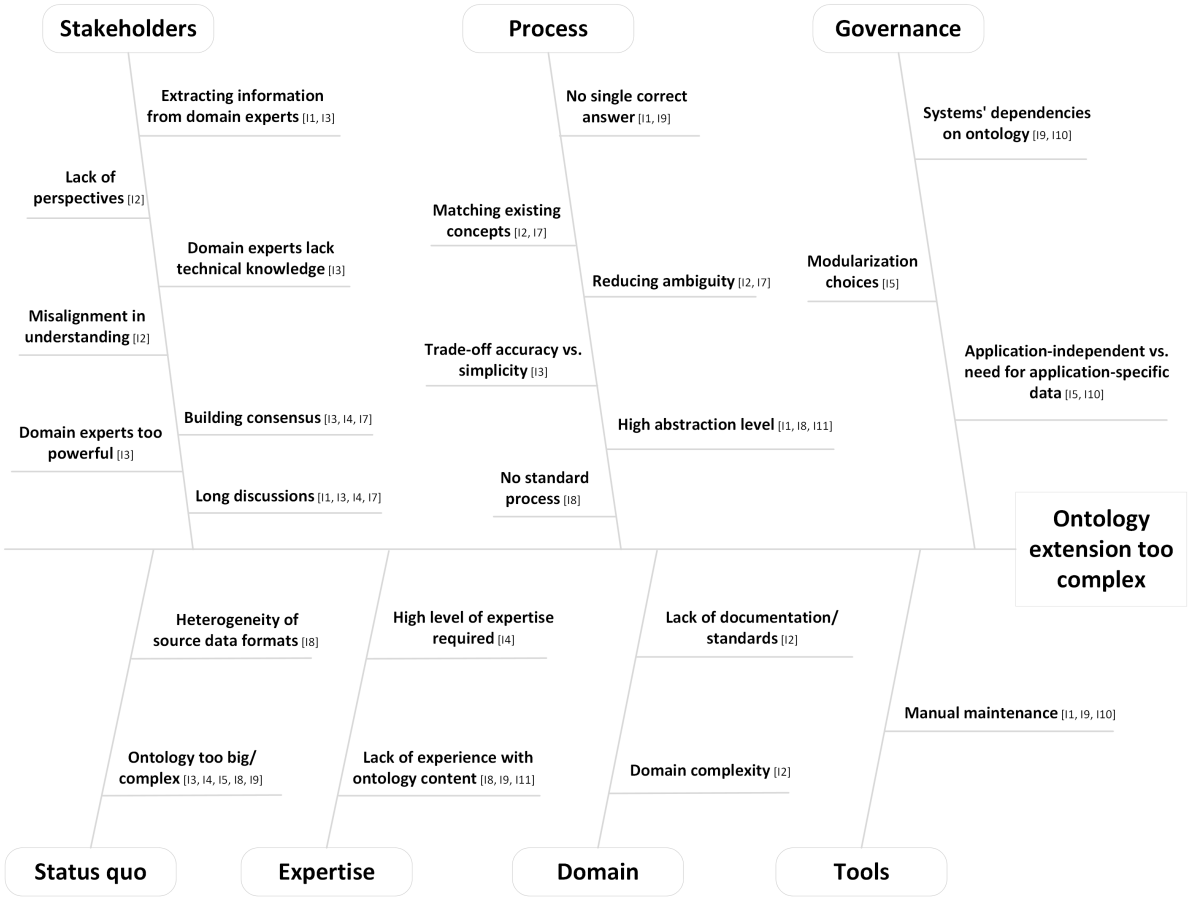


Figure 1: Ishikawa diagram mapping the problems causing the complexity in the ontology extension process. Based on the content analysis of the data collected through the interviews. Numbers in brackets refer to the interview(s) in which the problem was mentioned/discussed.

problematic in the field of OE, where logical consistency and factual accuracy are essential. Users may struggle to detect them, increasing the risk of flawed ontologies.

Environmental and ethical concerns were also raised by the interviewees, such as the substantial energy and water consumption associated with LLM training and maintenance. Additionally, the use of proprietary LLMs introduces risks related to data privacy and intellectual property, particularly when handling sensitive information. Finally, there is apprehension about the potential loss of the collaborative human process integral to OE. Figure 2 shows a visualization of the concerns mentioned by the interviewees about the application of LLMs to OE.

Despite the concerns, the interviewees saw several promising opportunities for enhancing ontology extension with LLMs. During the initial stages, they can offer creative and out-of-the-box suggestions for identifying domain concepts, especially when ontology engineers are unfamiliar with the domain or the required extension. LLMs can also aid in identifying concepts and relationships from unstructured text. Other opportunities are giving advice, SPARQL query generation, and repetitive tasks such as populating an ontology.

4.2. The human-LLM collaboration ontology extension workflow

Stemming from the problems in the ontology extension process and from the challenges and opportunities for LLMs to be introduced into this process, discussed above, a set of 22 high-level requirements for a human-LLM collaboration framework for ontology extension were identified. The complete list can be found in Table 2 in the Appendix. This output has been used to design a process framework for ontology extension using LLMs. In our process framework, we aim at a step-by-step interaction of the

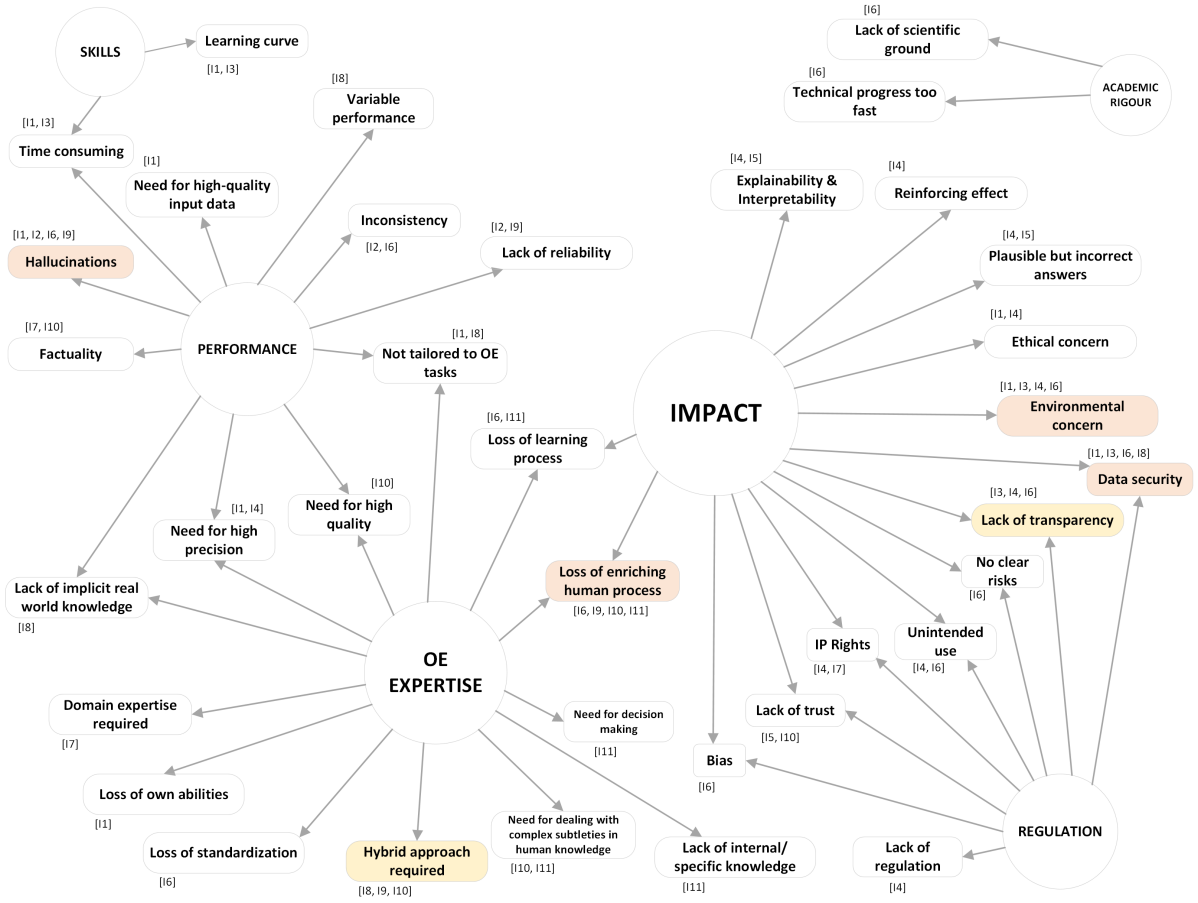


Figure 2: Concerns about the use of LLMs for OE. Visualization inspired by Network Theory, where the nodes with higher degree are bigger than the nodes with lower degree. The graph is based on the content analysis of the data collected through the interviews. Numbers in brackets refer to the interview(s) in which the problem was mentioned. The concerns that were mentioned by at least 4 interviewees are highlighted in orange, and the concerns that were mentioned by at least 3 are highlighted in yellow.

user with the LLM using simple prompts.

The human-LLM collaboration process framework for ontology extension (Figure 4) is an augmented version of the current process outlined above in Section 4.1.1. We have chosen the current ontology extension process as a template because it already maps the downstream ontology extension tasks in the different phases of the process, namely Preparation; Conceptualization; Implementation; Verification; Exploitation; and Validation. In addition, the flowchart format provides flexibility since the ontology engineer can choose the sequence of tasks (transparent rounded boxes in Figure 4) to be executed depending on the specific needs, represented as questions in the diagram (gray rectangular boxes in Figure 4). The preparation phase focuses on gathering documentation about the ontology to be extended and the domain. In the conceptualization phase, the flowchart outlines the steps to reuse concepts from other ontologies (left-hand side), and to build a sub-ontology guided by CQs to then align it with the ontology to be extended (right-hand side). In the implementation and validation phases, the ontology extension is coded in a formal language and validated by using the CQs, visualizations of the ontology, and existing OE tools such as OOPS! [42].

To map the downstream tasks in the ontology extension process to the NLP capabilities of LLMs, we used the categorization of NLP tasks proposed in [43]. Based on the definition of the ontology extension task and its mapping to an NLP task. As an example, “Define ontology extension modules” can be mapped to “Text Classification”, and “Formulate CQs” can be mapped to “Keyphrase Generation”. We indicate which tasks can be assisted by LLMs with a white star-shaped icon with red text and a purple

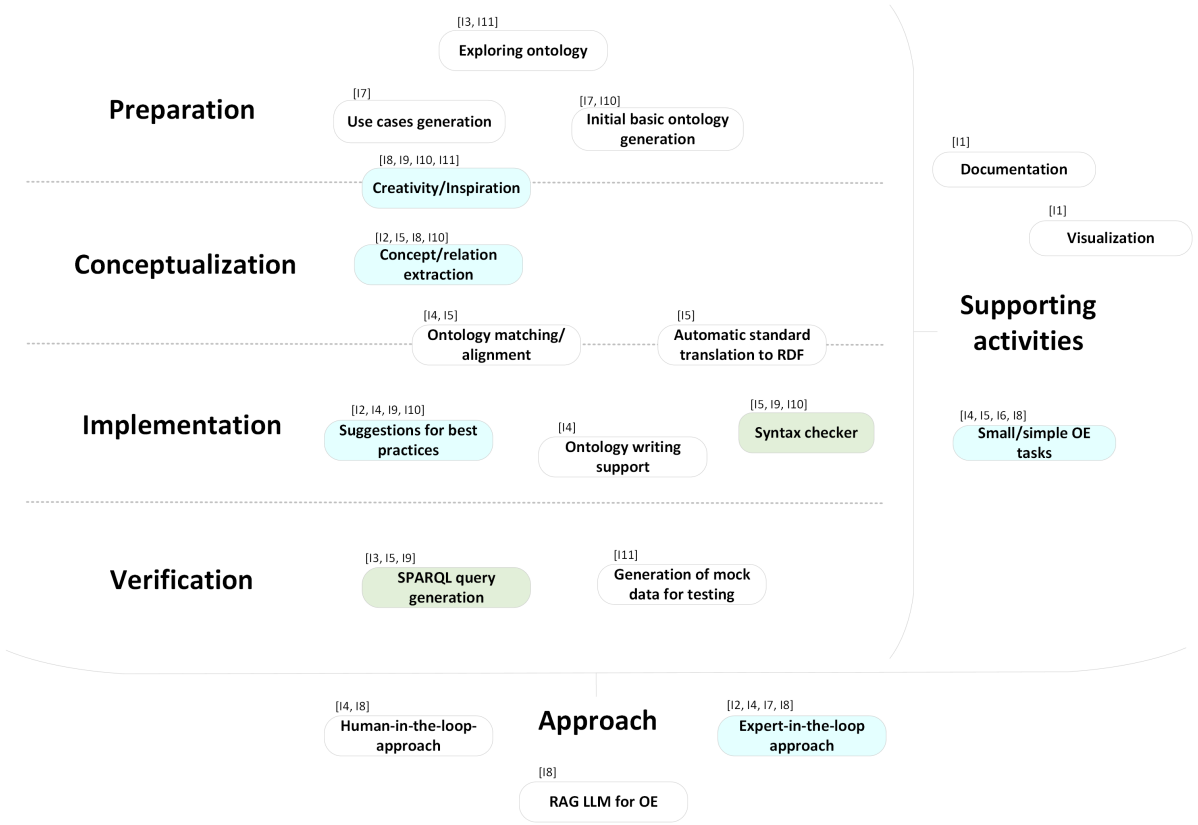


Figure 3: Opportunities for the use of LLMs for OE, mapped to the different OE phases. Figure based on the content analysis of the data collected through the interviews. Numbers in brackets refer to the interview(s) in which the idea was mentioned/discussed. The opportunities that were mentioned by at least 4 interviewees are highlighted in blue, and the opportunities that were mentioned by at least 3 are highlighted in green.

tag with a number in the flowchart diagram in Figure 4.

As seen the figure, most of the LLM-assisted tasks belong to the Preparation and Conceptualization phases. The interviews show that ontology engineers acknowledge the high complexity in these initial phases of the ontology extension process and the capabilities of LLMs to produce relevant ideas for augmenting human inspiration. For example, for task T-1.1 Research about the domain, LLMs can generate a well structured overview of the main aspects in a specific domain, including the relevant terminology [44]. In the Conceptualization phase, task T-2.5 consists of aligning the ontology extension with the ontology to be extended. As demonstrated by Amini et al. [36], LLMs can provide relevant suggestions for manual alignment by proposing 1-to-1 mappings.

As a result of the analysis, we propose a total of 15 ontology extension tasks could be facilitated by LLMs. In Figure 5 we provide a zoomed-in version of Figure 4 for better readability, specifically for phases 2) Conceptualization (Figure 5-a) and 4) Verification (Figure 5-b). For each task, we created a prompt template. As an example, Figure 6 shows (part of) a prompt template for task T-1.6 Create glossary of terms. All the prompt templates can be found in GitLab[5].

Beyond the NLP capabilities of LLMs, we also examined recent publications such as the OntoChat framework [18] and the NeOn-GPT workflow for ontology modeling [16], both with interesting results that we have incorporated to our design. OntoChat can be used within different phases, specifically in tasks T-1.5, T-2.1, and T-4.2 to define business scenarios, formulate CQs, and verify whether the ontology extension can answer the CQs, respectively. On the other hand, the authors of the NeOn-GPT workflow demonstrate the potential of using GPT in combination with OOPS! [42] to detect and fix structural and syntax errors in the ontology. Following their approach, the LLM could be used in

[5]https://gitlab.com/eswc2025/ontology-extension-with-llms/-/tree/main/Prompt_Templates

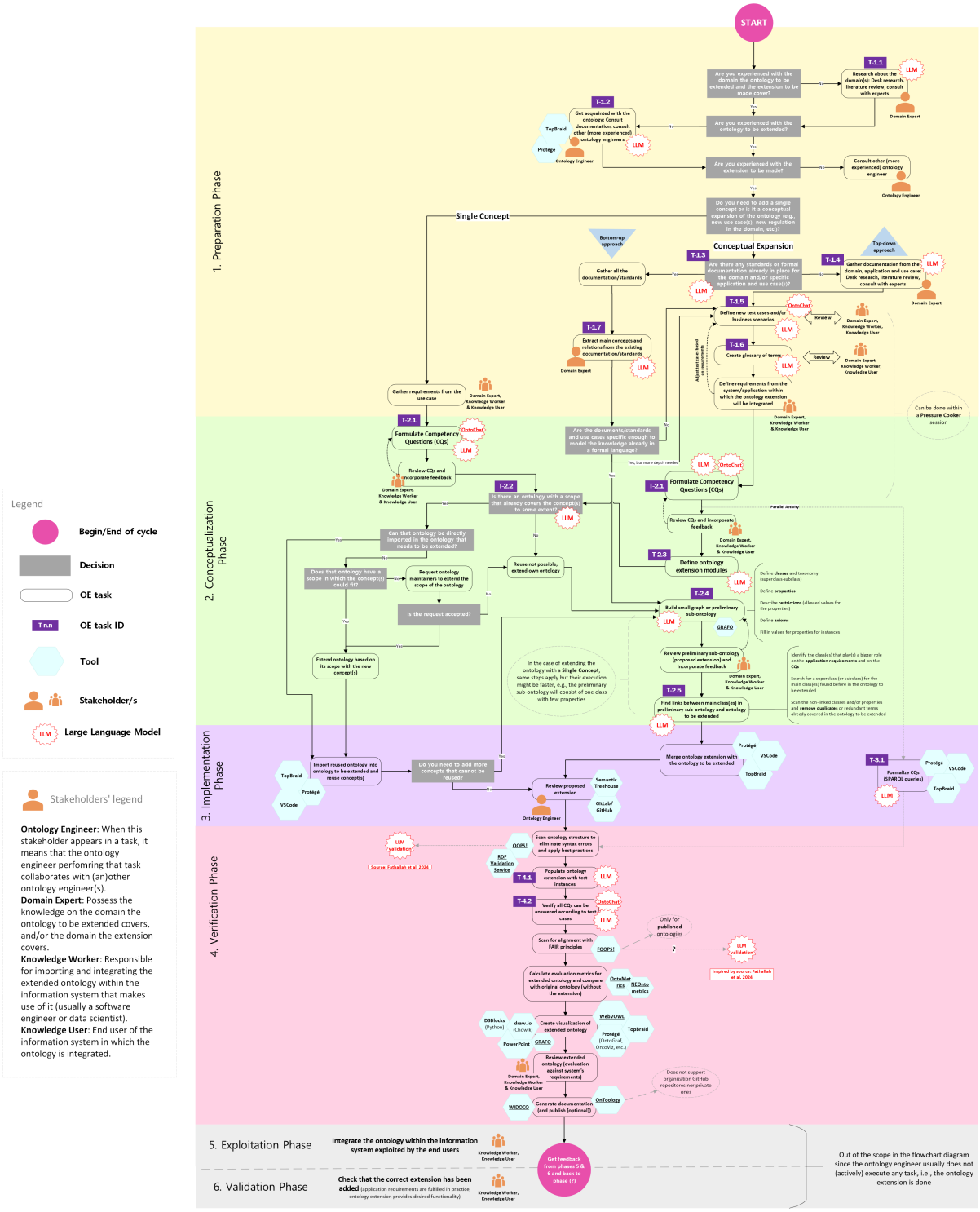
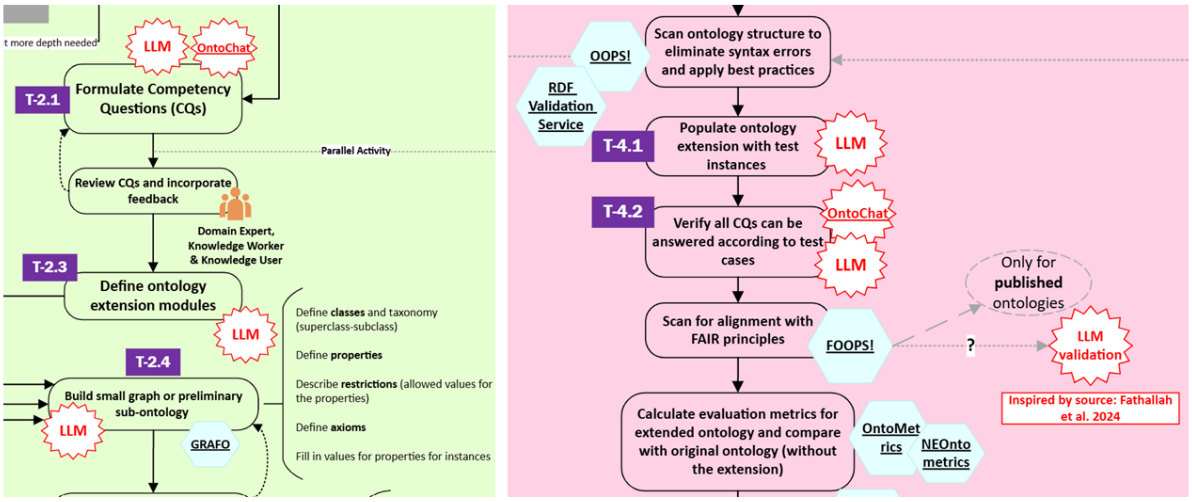


Figure 4: Design for the ontology extension process framework for human-LLM collaboration.

combination with the tool FOOPS! [45] to align the ontology with the FAIR principles.

In addition to OOPS! [42] and FOOPS! [45], we integrated other OE tools in the ontology extension framework for human-LLM collaboration that are not currently used by the ontology engineers that we interviewed. These are Grafo[6]; OntoEditor [46]; OntoMetrics [47] and its updated version NEOntometrics [48]; WIDOCO [49]; and OnToology [50]. We selected these tools because they are still available and maintained [December 2024]. Although these tools will not eliminate the complexity inherent in

[6]https://gra.fo/



a) b)

Figure 5: Zoomed in excerpt from the human-LLM collaboration process framework in Figure 4. a) Conceptualization phase, LLM assisted tasks T-2.1, T-2.3, and T-2.4. b) Verification phase, LLM assisted tasks T-4.1 and T-4.2. Question mark indicates that the task (LLM validation of fair principles using FOOPS!) has not been tested. Blue hexagon-shaped boxes contain OE tools suggestions for the task.

- **Upload documentation or text about the domain or extension use case description** (if not uploaded before).
- **Upload test cases/business scenarios.**
- **Prompt 1**: *From the documentation provided in file "{name_of_file_provided}", I want you to provide a list of the most important concepts, i.e., the concepts that are mentioned more frequently. For each one, identify if the concept is a noun or a verb and the number of times it appears in the provided documents (an integer). Make the list as complete as possible. Your answer must be in the following format:*

```
1. <Concept_1 >: (<noun/verb >). (<# of times it appears >).
```

- **Prompt 2**: *Now, from the documentation provided before in file "{name_of_file_provided}", can you provide a definition for each concept you identified? The format of your answer must be a list in which each row looks like this:*

```
1. <Concept_1 >: <Definition >.
```

- **Prompt 3**: *From the list provided {before/below (if reviewed list, prompt new list)}, now indicate possible synonyms for each concept (if any) according to their definition in the domain. Do not create any new concept, stick to the ones provided in the list. Provide a new list with the following format:*

Figure 6: Excerpt of prompt template for task T-1.6 Create glossary of terms.

the ontology extension process, they can assist with certain downstream tasks.

5. Evaluation

To evaluate the process framework presented in Figure 4, we executed the different LLM-assisted steps and compared them against a manually created ground truth. In this section, we discuss the manually created extension, in Section 5.1. Next, we present highlights from the human-LLM collaboration process framework in Section 5.2. All the results from the demonstration and evaluation, including the

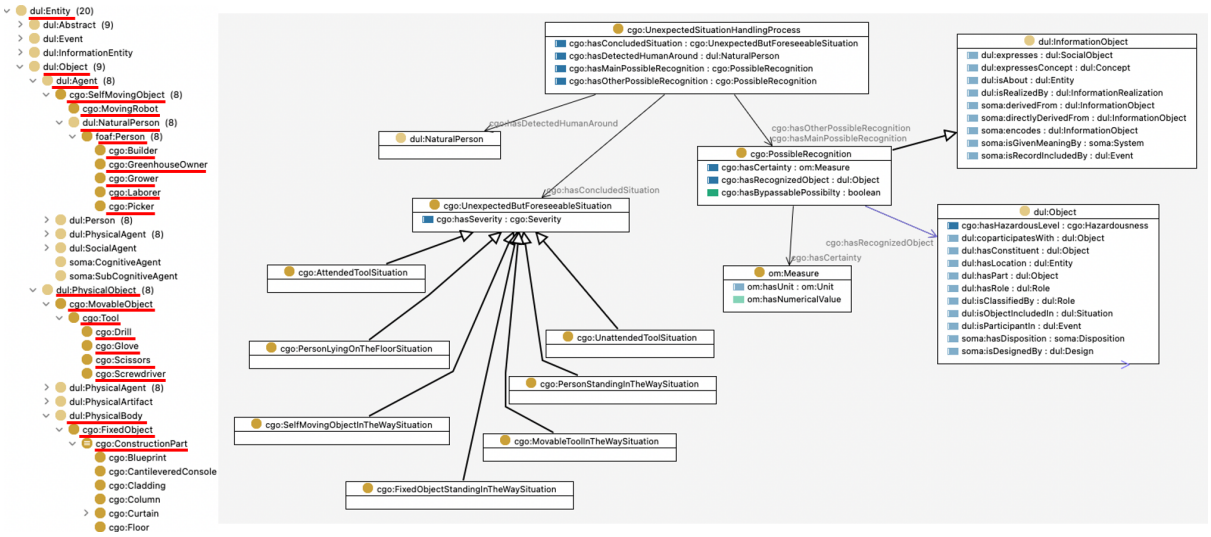


Figure 7: The SENS extension to the CGO, the gold standard. Object hierarchy (left) and graph visualization (right). Highlighted in red some of the main concepts added for the SENS extension.

generated code, can be found in GitLab[7].

5.1. The gold standard

With the goal of enabling semantic communication and interoperability between robots from different manufacturers in the greenhouse, the Common Greenhouse Ontology (CGO)[8] [33] was extended with new concepts and relations within the Semantic Explanation and Navigation System (SENS)[9] use case. The CGO is a public domain ontology focused on high-tech greenhouse infrastructure and the systems measuring the necessary data to effectively monitor and control the crops' growth inside the greenhouse, such as climate data [33, 34]. Figure 7 illustrates the manual SENS extension to the CGO developed and used here as our gold standard. The visualization has been generated with TopBraid, using the code for the use case SENS in the public GitLab repository of the CGO[10].

SENS assumes that the use of autonomous systems in high-tech greenhouses will grow in the near future. Consequently, heterogeneous autonomous systems (modeled as "cgo:MovingRobot") will coexist in the greenhouse, together with human workers, such as growers, pickers, technicians, or managers (modeled under "dul:NaturalPerson"). The greenhouse is a tight space composed of narrow paths and dense vegetation in which human workers and autonomous systems or robots must perform different tasks concerning the crops and use specific materials and tools (modeled under "cgo:Tool", extended from "dul:PhysicalObject"). These objects can be in motion (e.g., carts), or could potentially be hazardous in specific situations (e.g., weeding knives or scissors). Currently, robots can detect obstacles obstructing their way, but will only stop without notifying the human operator. This may lead to unnecessary idle times and delays. The goal of SENS is to improve this scenario concerning foreseeable but unexpected situations (modeled taxonomy under "cgo:UnexpectedButForeseeableSituation") by enabling the robots to semantically communicate and explain to the human operator the obstacle encountered and the level of urgency of the situation ("cgo:hasSeverity" and "cgo:Severity"). With this information, the human operator can locate the robot in the greenhouse, judge the situation, and act accordingly (e.g., removing the obstacle from the robot's way).

The manual extension to the CGO (i.e., without using LLMs) was conceptualized in 2023 by an experienced ontology engineer. The SENS extension development was based on the requirements of a dashboard in which notifications could be shown to the user, based on the knowledge about types of

[7] https://gitlab.com/eswc2025/ontology-extension-with-llms/-/tree/main/Demonstration_and_Evaluation?ref_type=heads
[8] https://gitlab.com/ddings/common-greenhouse-ontology
[9] https://appl-ai-tno.nl/projects/sens
[10] https://gitlab.com/ddings/common-greenhouse-ontology/-/tree/sens?ref_type=heads

obstacles found in the greenhouse and types of situations and actions related to the characteristics of the obstacles or objects. The resulting extension is shown in Figure 7. The ontology engineer executed the following tasks: find reusable existing ontologies, define some competency questions, define new concepts/properties, find the best spot to place them in the existing CGO and add them using Topbraid Composer. From four ontologies identified as potentially reusable for the SENS use case extension, two ontologies were selected and reused: the SOMA ontology [51] and the Dolce/DUL ontology [52]. The CGO was extended by adding the concepts and relations for SENS one by one to the ontology. The process of manually extending the CGO with the SENS use case took approximately 40 hours.

5.2. Using the process framework to generate SENS

To demonstrate and evaluate the human-LLM collaboration framework for ontology extension prototype we performed a walk-through of all the tasks in the process framework (Figure 4) that can be assisted by LLMs. All the prompt templates have been adjusted to the SENS use case, replacing the placeholders by the corresponding information concerning the CGO and the greenhouse domain, and using the description of the use case SENS. We opted to use a custom GPT Assistant[11] using OpenAI's GPT-4 Omni model[12]. In its file store, we included a text file containing relevant information about the CGO (text of 639 words in PDF format), taken from its public GitLab repository, and a text file containing the CGO code (the full ontology) formatted as a Turtle triples file. We used a version of the CGO code before the implementation of SENS (i.e., the CGO code uploaded to the file store of the GPT Assistant does not contain any data related to the SENS use case).

Both manual and automated ontology extensions have distinct strengths and weaknesses (see Table 5 in the Appendix). The manual approach achieves a richer taxonomy and better reuse of other ontologies compared to the LLM-assisted approach, where the taxonomy is simpler and GPT hallucinates ontologies to reuse. However, the extension generated using the framework proposes an original model for the robot's decisions when encountering obstacles not considered in the gold standard's development. For each output of the tasks executed using the LLM (in this case the GPT Assistant), we carefully observed the results, reflecting on the output of the LLM and focusing on the correctness and usefulness of the task for the ontology engineer. All the inputs prompted to the LLM and all the outputs generated are publicly available in our GitLab repository[13]. The main insights are:

- **Preparation phase – Getting acquainted with the domain, the ontology to be extended, and the extension**: Tasks T-1.1, T-1.2, and T-1.3 produced relevant results that can be used by the ontology engineer to get acquainted with the domain, the ontology to be extended and the ontology extension. The outputs can serve as inspiration to the ontology engineer, but the information must be reviewed and checked, for example by opening the ontology file in Protégé or TopBraid. The LLM can complement the information provided by these tools, though not replace them.
- **Preparation phase – Gathering existing standards**: The output of task T-1.4 was fully hallucinated. Thus, this indicates that the ontology engineer should use conventional search engines instead of an LLM for this task.
- **Preparation phase – Defining business scenarios, creating a glossary of terms, and extracting concepts and relations from existing standards**: tasks T-1.5, T-1.6, and T-1.7 produced relevant suggestions with few-shot prompting and by making the instructions in the prompts very specific.
- **Conceptualization phase – Formulating Competency Questions**: Though also highly dependent on the quality of the documentation of the use case provided, the output of task T-2.1 was surprisingly relevant. We provided the complete list of CQs generated by the LLM in Table 3 of Appendix A.2. The quality achieved was not expected after examining the results of previous

[11] https://platform.openai.com/docs/assistants/overview
[12] https://openai.com/index/hello-gpt-4o/
[13] https://gitlab.com/eswc2025/ontology-extension-with-llms/-/tree/main/Demonstration_and_Evaluation?ref_type=heads

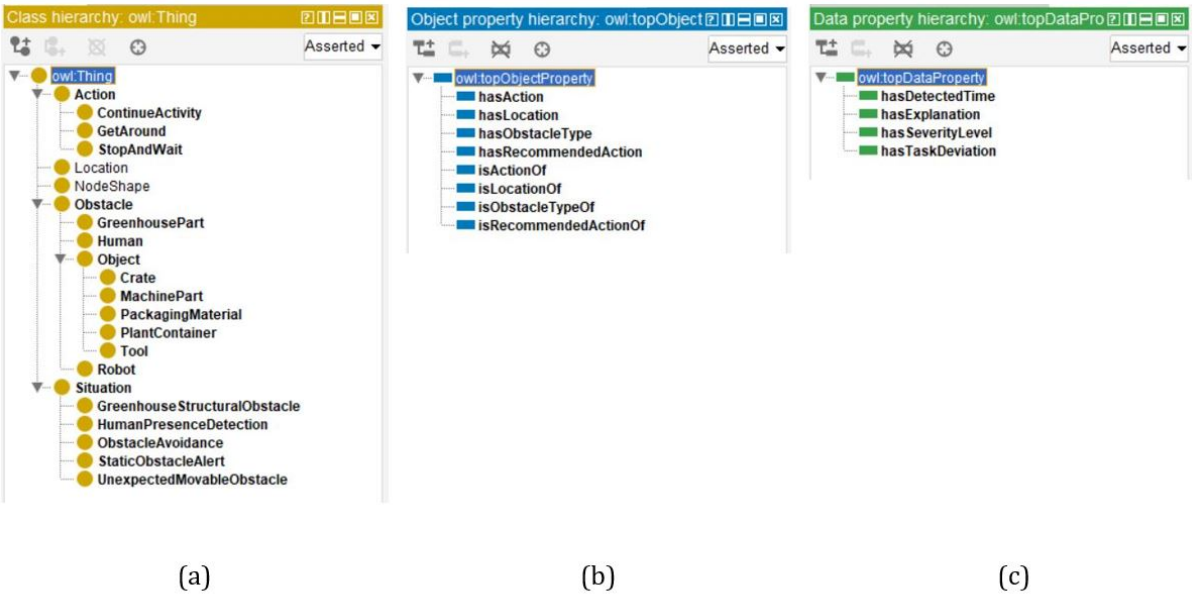


Figure 8: Output of task T-2.4 – Final version of the ontology extension (concepts to be added to the CGO) generated using the human LLM collaboration process framework prototype (after additional prompts) – a) Classes; b) Object properties; and c) Data properties. Visualized in Protégé.

research on the topic [23]. In this case, the CQs produced were similar to the ones formulated in the gold standard (e.g., *CQ1: What type of obstacle has the robot detected?*, *CQ2: Where is the robot currently located?*) and some of them provided relevant suggestions that were not thought of in the gold standard but that could be added (e.g., *CQ10: How much time did the robot take to avoid the obstacle?*, *CQ13: How many obstacles have been detected within a specified time-frame*?).

- **Conceptualization phase – Reusing existing ontologies and defining modules**: The output to task T-2.2 included 1 correct suggestion of an ontology that could be reused for the use case SENS, from the 2 ontologies reused in the gold standard (in total 4 were proposed). The output to task T-2.3 produced coherent suggestions to modularize the ontology, though it is not clear if it is useful in reality, since the ontology extension is small in this case.
- **Conceptualization phase – Building the ontology extension and aligning the extension with the ontology to be extended**: The output of task T-2.4 needed to combine few-shot prompting to the prompting chaining technique proposed in order to add depth to the ontology (for example, to add sub-classes to the class Obstacle and to the class Situation), and to apply several corrections. But overall, the code provided by GPT in Turtle/OWL syntax was syntactically correct and the model was able to identify mistakes and correct them. It is worth mentioning that GPT established severity levels for the situations according to their characteristics and the impact on the functioning of the robot, without specifying this in any prompt. The reasoning by GPT is logical and similar to the gold standard. Thus, this task is useful if the ontology engineer wants to use GPT to obtain an initial version of an ontology without having to write a single line of code or without having to do it manually in Protégé. For an inexperienced ontology engineer, this task might be much more useful, providing a solid starting point while learning good OE practices. The final ontology resulting from this task can be seen below in Figure 8. Some prompts were inspired by the work of Fathallah et al. [16]. Following the approach of Amini et al. [36], the suggestions for aligning the ontology extension with the ontology to be extended given in task T-2.5 are relevant and provide additional insights compared to the gold standard.
- **Implementation phase – Formalizing CQs into SPARQL queries**: The output of task T-3.1 was unexpectedly high quality with zero-shot prompting. Though the potential of using LLMs for this task has been demonstrated, the performance of LLMs for this task is claimed to be "unstable"

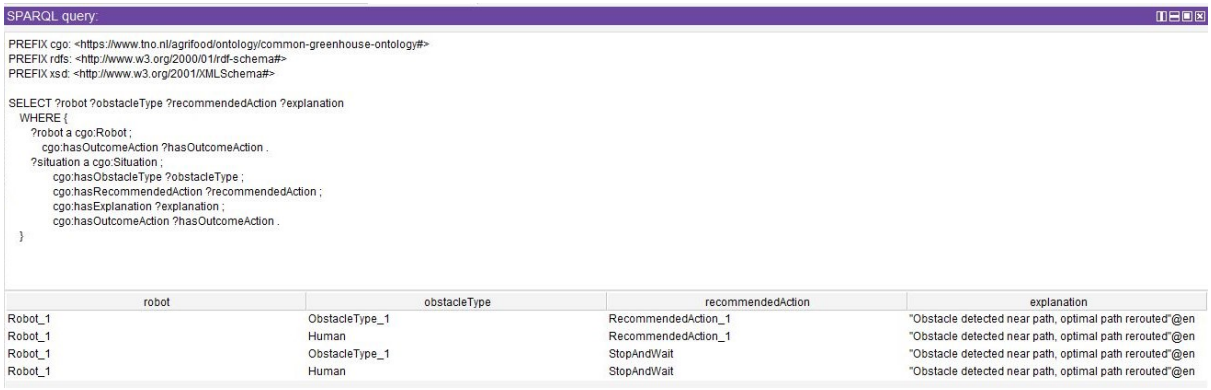


Figure 9: Output of Task 4.1 – Competency Question 6: Why has the robot made the specific decision for obstacle avoidance?. The SPARQL query previously generated within the task 3.1 was executed after populating the ontology with individuals. Visualized in Protégé.

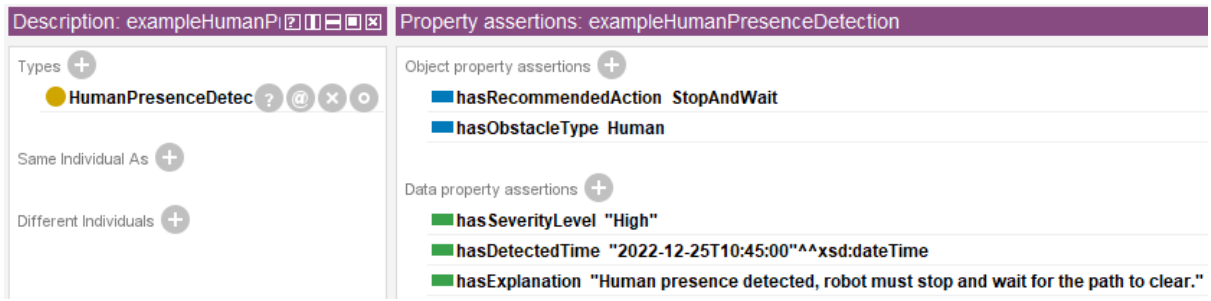


Figure 10: Output of Task 4.1 – Individual: "exampleHumanPresenceDetection". Visualized in Protégé.

[53]. In this test, from the 16 CQs provided to GPT, all were syntactically correct when executed in Protégé and 8 gave results. Some SPARQL queries produced revealed the reasoning capabilities of GPT, such as *CQ6: Why has the robot made the specific decision for obstacle avoidance?*, in which it is not explicit how the robot makes a decision (see Figure 9). As shown in Figure 9, the SPARQL query produced shows that GPT correctly identified that to answer that CQ, information about the outcome of the situation, the obstacle type, the recommended action, and the explanation was needed. Though the SPARQL queries produced by GPT might need some manual post-processing, perhaps to simplify them, this output gives a solid start to ontology engineers and might be especially relevant for domain experts with scarce knowledge on SPARQL query generation.

- **Verification phase – Populating ontology and verifying CQs**: With task T-4.1 the ontology extension was populated with relevant individuals (see Figure 10). As a result, 14 out of the 16 SPARQL queries generated previously gave results when executed in Protégé. This task can be especially useful since it can fully eliminate the cumbersome process of having to create manually a lot of individuals, as discussed with Interviewee 11). The output of task T-4.2, provided in Table 4 of Appendix A.2, inspired by the work of Zhang et al. [18], is almost fully correct, with 14 out of 16 CQs correctly identified. The potential of this task relies on the capacity of the user to spot the mistakes and judge the output, but it can provide a solid starting point and serve as a guide to less experienced ontology engineers. The highest benefit of this task is that the ontology extension can be verified without having to use SPARQL queries, which could make the task especially useful for domain experts (who might not know how to write SPARQL queries).

After executing and assessing the results of these ontology extension activities, as guided by the process framework (Figure 4) we can conclude that, overall, the generated SENS extension as a result of the human-LLM collaboration process was correct and useful. The major issues are lack of depth and complexity (a human can generate more sub-classes and more and more sophisticated axioms) and the need to integrate the ontology to be extended with the ontology extension manually. The proposed

process framework can be specially useful for beginner ontology engineers who are familiar with the basic concepts in ontology engineering (including domain experts with little technical experience in OE). Our framework supports them by providing a step-by-step guide based on best practices and automating a set of tasks so that the user has a starting point to further develop the extension.

It is important to highlight that, during the execution of the LLM-assisted tasks in the ontology extension process, the LLM hallucinated in 2 tasks: when asked about existing standards covering the extension use case; and when asked about existing ontologies to be reused. In our proposed framework, hallucinations are managed manually by the user of the framework. The user must be able to spot the hallucinations and manually correct the answer of the LLM, before continuing to the next task in the process.

The generation of the SENS extension using the process framework and the GPT Assistant took approximately 16 hours of a beginner ontology engineer. As previously mentioned, the manual SENS extension (the gold standard) was developed in approximately 40 hours by an experienced ontology engineer

6. Conclusion and Future Work

In this paper, we present a human-LLM collaboration process framework for ontology extension. This framework supports the human ontology engineer and/or domain expert with an LLM in multiple tasks. To evaluate its qualitative performance, we applied the framework to extend an existing greenhouse ontology with new concepts and properties of the domain and compared the result to a manually generated extension, our gold standard. Furthermore, we evaluated each task's output to determine how effectively an LLM reduces manual effort and enhances human creativity.

The main conclusion of our evaluation is that LLMs are a useful tool for the human ontology engineer to (1) get inspiration on where and how to add new concepts and properties, (2) deal with complex syntax definitions and repetitive tasks, and (3) verify whether the extended ontology conforms to the initially defined requirements and competency questions. Our experiments with the greenhouse ontology show that the proposed framework can lower the entry barriers to the field of ontology engineering, because it guides the ontology engineer and reduces manual effort in some of the tasks. However, due to the problematic hallucinations and because not all ontology engineers are familiar with the use of LLMs, additional training including critical thinking would be necessary for effective interaction of the user of the process framework with the LLM.

We noticed that important preconditions for successful usage of LLMs are (1) specific fine-tuning of the prompt inputs to the LLM, (2) a user-friendly interface with the LLM that provides task-specific support, and (3) last but not least, expert involvement to check the LLM output for correctness and completeness, and to mitigate hallucinations. Furthermore, our interviews demonstrate that more general aspects, that we did not study, such as transparency, trustworthiness, security and environmental impact, should be taken into account when deciding to use an LLM. Previous work on this topic shows that the full automation of the ontology engineering process is currently not possible due to the limitations of LLMs. Our work further indicates that ontology engineers do not favor full automation, as they view this process as inherently human and highly enriching.

In this research, we manually evaluated the process framework design using only the common greenhouse ontology. Since our framework is applicable to any ontology, we plan to further assess the approach with other ontologies across various domains to enhance the generalizability of our results. LLMs may lack the specialized knowledge required for fields like biomedical or legal domains, and thus more expert intervention might be needed when extending ontologies in these domains using our framework. In addition, future work on this topic includes a more extensive evaluation of the framework by ontology engineers with different levels of experience; a study the potential of applying fine-tuned open-source LLMs for specific and smaller tasks such as creating a glossary of terms from unstructured text sources; and the development a user-friendly ontology engineering tool based in our framework, to seamlessly integrate LLMs within the ontology engineering toolkit.

Acknowledgments

This research was supported by the project NXTGEN Hightech. Our gratitude goes to the Common Greenhouse Ontology (CGO) team and the Semantic Explanation and Navigation System (SENS) project for providing us with the use case and material to conduct our research. Profound gratitude is extended to the participants in the interviews. The authors have no competing interests to declare that are relevant to the content of this article.

Declaration on Generative AI

During the preparation of this work, the author(s) used Microsoft Copilot in order to: Grammar, spelling check, and improving the readability of some sentences. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] L. Obrst, Ontologies for semantically interoperable systems, in: Proceedings of the twelfth international conference on Information and knowledge management, CIKM '03, Association for Computing Machinery, New York, NY, USA, 2003, pp. 366–369. doi:10.1145/956863.956932.

[2] T. Tran, H. Lewen, P. Haase, On the Role and Application of Ontologies in Information Systems, in: 2007 IEEE International Conference on Research, Innovation and Vision for the Future, 2007, pp. 14–21. doi:10.1109/RIVF.2007.369130.

[3] M. Funk, S. Hosemann, J. C. Jung, C. Lutz, Towards Ontology Construction with Language Models, arXiv preprint arXiv:2309.09898 (2023). doi:10.48550/arXiv.2309.09898.

[4] A. Beerli, R. Indergand, J. S. Kunz, The supply of foreign talent: how skill-biased technology drives the location choice and skills of new immigrants, Journal of Population Economics 36 (2023) 681–718. doi:10.1007/s00148-022-00892-3.

[5] F. Neuhaus, Ontologies in the era of large language models – a perspective, Applied Ontology 18 (2023) 399–407. doi:10.3233/AO-230072, publisher: IOS Press.

[6] A. Oba, I. Paik, A. Kuwana, Automatic Classification for Ontology Generation by Pretrained Language Model, in: H. Fujita, A. Selamat, J. C.-W. Lin, M. Ali (Eds.), Advances and Trends in Artificial Intelligence. Artificial Intelligence Practices, Lecture Notes in Computer Science, Springer International Publishing, Cham, 2021, pp. 210–221. doi:10.1007/978-3-030-79457-6_18.

[7] H. Babaei Giglou, J. D'Souza, S. Auer, LLMs4OL: Large Language Models for Ontology Learning, in: T. R. Payne, V. Presutti, G. Qi, M. Poveda-Villalón, G. Stoilos, L. Hollink, Z. Kaoudi, G. Cheng, J. Li (Eds.), The Semantic Web – ISWC 2023, Springer Nature Switzerland, Cham, 2023, pp. 408–427. doi:10.1007/978-3-031-47240-4_22.

[8] P. Mateiu, A. Groza, Ontology engineering with Large Language Models, IEEE Computer Society, 2023, pp. 226–229. doi:10.1109/SYNASC61333.2023.00038.

[9] R. Studer, V. R. Benjamins, D. Fensel, Knowledge engineering: Principles and methods, Data & knowledge engineering 25 (1998) 161–197.

[10] P. Buitelaar, P. Cimiano, B. Magnini, Ontology learning from text: methods, evaluation and applications, volume 123 of Frontiers in Artificial Intelligence and Applications, IOS press, 2005.

[11] M. de Boer, J. P. Verhoosel, Towards data-driven ontologies: a filtering approach using keywords and natural language constructs, in: Proceedings of the Twelfth Language Resources and Evaluation Conference, 2020, pp. 2285–2292.

[12] M. N. Asim, M. Wasim, M. U. G. Khan, W. Mahmood, H. M. Abbasi, A survey of ontology learning techniques and applications, Database, The Journal of Biological Databases and Curation 2018 (2018) 1–24. doi:10.1093/database/bay101.

[13] A. C. Khadir, H. Aliane, A. Guessoum, Ontology learning: Grand tour and challenges, Computer Science Review 39 (2021) 100339.

[14] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, D. Amodei, Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.

[15] S. Hao, B. Tan, K. Tang, H. Zhang, E. P. Xing, Z. Hu, Bertnet: Harvesting knowledge graphs from pretrained language models, arXiv preprint arXiv:2206.14268 (2022).

[16] N. Fathallah, A. Das, S. De Giorgis, A. Poltronieri, P. Haase, L. Kovriguina, NeOn-GPT: A Large Language Model-Powered Pipeline for Ontology Learning, in: European Semantic Web Conference (ESWC) 2024, Hersonissos, Greece, 2024. Presented in the Special Track on Large Language Models for Knowledge Engineering.

[17] M. J. Saeedizade, E. Blomqvist, Navigating Ontology Development with Large Language Models, in: A. Meroño Peñuela, A. Dimou, R. Troncy, O. Hartig, M. Acosta, M. Alam, H. Paulheim, P. Lisena (Eds.), The Semantic Web, Springer Nature Switzerland, Cham, 2024, pp. 143–161. doi:10.1007/978-3-031-60626-7_8.

[18] B. Zhang, V. A. Carriero, K. Schreiberhuber, S. Tsaneva, L. S. González, J. Kim, J. de Berardinis, OntoChat: a Framework for Conversational Ontology Engineering using Language Models, in: European Semantic Web Conference (ESWC) 2024, Hersonissos, Greece, 2024. Presented in the Special Track on Large Language Models for Knowledge Engineering.

[19] M. Trajanoska, R. Stojanov, D. Trajanov, Enhancing Knowledge Graph Construction Using Large Language Models, arXiv preprint arXiv:2305.04676 (2023). doi:10.48550/arXiv.2305.04676.

[20] A. S. Lippolis, M. Ceriani, S. Zuppiroli, A. G. Nuzzolese, Ontogenia: Ontology Generation with Metacognitive Prompting in Large Language Models, in: A. Meroño Peñuela, O. Corcho, P. Groth, E. Simperl, V. Tamma, A. G. Nuzzolese, M. Poveda-Villalón, M. Sabou, V. Presutti, I. Celino, A. Revenko, J. Raad, B. Sartini, P. Lisena (Eds.), The Semantic Web: ESWC 2024 Satellite Events, Springer Nature Switzerland, Cham, 2025, pp. 259–265. doi:10.1007/978-3-031-78952-6_38.

[21] R. M. Bakker, D. L. Di Scala, From text to knowledge graph: Comparing relation extraction methods in a practical context, in: First International Workshop on Generative Neuro-Symbolic AI, co-located with ESWC, 2024.

[22] R. M. Bakker, D. L. Di Scala, M. H. T. de Boer, Ontology Learning from Text: an Analysis on LLM Performance, in: Proceedings of the 3rd NLP4KGC International Workshop on Natural Language Processing for Knowledge Graph Creation, co-located with Semantics 2024, Amsterdam, Netherlands, 2024.

[23] Y. Rebboud, L. Tailhardat, P. Lisena, R. Troncy, Can LLMs Generate Competency Questions?, in: ESWC 2024, Extended Semantic Web Conference, Hersonissos, Greece, 2024. URL: https://hal.science/hal-04564055.

[24] R. Alharbi, V. Tamma, F. Grasso, T. R. Payne, The Role of Generative AI in Competency Question Retrofitting, in: A. Meroño Peñuela, O. Corcho, P. Groth, E. Simperl, V. Tamma, A. G. Nuzzolese, M. Poveda-Villalón, M. Sabou, V. Presutti, I. Celino, A. Revenko, J. Raad, B. Sartini, P. Lisena (Eds.), The Semantic Web: ESWC 2024 Satellite Events, Springer Nature Switzerland, Cham, 2025, pp. 3–13. doi:10.1007/978-3-031-78952-6_1.

[25] Y. Zhu, X. Wang, J. Chen, S. Qiao, Y. Ou, Y. Yao, S. Deng, H. Chen, N. Zhang, LLMs for Knowledge Graph Construction and Reasoning: Recent Capabilities and Future Opportunities, arXiv prepprint arXiv:2305.13168 (2024). doi:10.48550/arXiv.2305.13168.

[26] X. Wei, X. Cui, N. Cheng, X. Wang, X. Zhang, S. Huang, P. Xie, J. Xu, Y. Chen, M. Zhang, Y. Jiang, W. Han, Zero-Shot Information Extraction via Chatting with ChatGPT, arXiv preprint arXiv:2302.10205 (2023). doi:10.48550/arXiv.2302.10205.

[27] L.-P. Meyer, C. Stadler, J. Frey, N. Radtke, K. Junghanns, R. Meissner, G. Dziwis, K. Bulert, M. Martin, LLM-assisted Knowledge Graph Engineering: Experiments with ChatGPT, in: C. Zinke-Wehlmann, J. Friedrich (Eds.), First Working Conference on Artificial Intelligence Development for a Resilient and Sustainable Tomorrow, Springer Fachmedien, Wiesbaden, 2024, pp. 103–115. doi:10.1007/978-3-658-43705-3_8.

[28] M. C. Suárez-Figueroa, A. Gómez-Pérez, M. Fernández-López, The NeOn Methodology framework:

A scenario-based methodology for ontology development, Applied Ontology 10 (2015) 107–145. doi:10.3233/AO-150145, publisher: IOS Press.
[29] K. Peffers, T. Tuunanen, M. A. Rothenberger, S. Chatterjee, A Design Science Research Methodology for Information Systems Research, Journal of Management Information Systems 24 (2007) 45–77. doi:10.2753/MIS0742-1222240302.
[30] A. Hevner, S. Chatterjee, Design Science Research in Information Systems, in: A. Hevner, S. Chatterjee (Eds.), Design Research in Information Systems: Theory and Practice, Springer US, Boston, MA, 2010, pp. 9–22. doi:10.1007/978-1-4419-5653-8_2.
[31] P. Johannesson, E. Perjons, An Introduction to Design Science, Springer International Publishing, Cham, 2021. doi:10.1007/978-3-030-78132-3.
[32] R. Falbo, SABiO: Systematic approach for building ontologies, in: 1st Joint Workshop ONTO.COM/ODISE on Ontologies in Conceptual Modeling and Information Systems Engineering, Co-located with FOIS, CEUR, Rio de Janeiro, RJ, Brazil, 2014. URL: https://www.semanticscholar.org/paper/SABiO%3A-Systematic-Approach-for-Building-Ontologies-Falbo/2660676b0d784d46d3c407793c0cee0c10e76c5b.
[33] R. M. Bakker, R. van Drie, C. Bouter, S. van Leeuwen, L. van Rooijen, J. Top, The Common Greenhouse Ontology: An Ontology Describing Components, Properties, and Measurements inside the Greenhouse, Engineering Proceedings 9 (2021) 27. doi:10.3390/engproc2021009027, publisher: Multidisciplinary Digital Publishing Institute.
[34] J. Verhoosel, B. Nouwt, R. M. Bakker, A. Sapounas, B. Slager, Digitizing Agriculture: A Datahub for Semantic Interoperability in Data-Driven Integrated Greenhouse Systems, ResearchGate, Rhodes Island, Greece, 2023.
[35] S. Hertling, H. Paulheim, OLaLa: Ontology Matching with Large Language Models, in: Proceedings of the 12th Knowledge Capture Conference 2023, K-CAP '23, Association for Computing Machinery, New York, NY, USA, 2023, pp. 131–139. doi:10.1145/3587259.3627571.
[36] R. Amini, S. S. Norouzi, P. Hitzler, R. Amini, Towards Complex Ontology Alignment using Large Language Models, arXiv preprint arXiv:2404.10329 (2024). doi:10.48550/arXiv.2404.10329.
[37] N. F. Noy, D. L. McGuinness, Ontology Development 101: A Guide to Creating Your First Ontology (2001). URL: http://protege.stanford.edu/publications/ontology_development/ontology101.pdf.
[38] M. McDaniel, V. C. Storey, V. Sugumaran, Assessing the quality of domain ontologies: Metrics and an automated ranking system, Data & Knowledge Engineering 115 (2018) 32–47. doi:10.1016/j.datak.2018.02.001.
[39] R. M. Bakker, M. H. T. De Boer, Dynamic Knowledge Graph Evaluation (2024). URL: https://www.techrxiv.org/users/791451/articles/1070833-dynamic-knowledge-graph-evaluation.
[40] K. I. Kotis, G. A. Vouros, D. Spiliotopoulos, Ontology engineering methodologies for the evolution of living and reused ontologies: status, trends, findings and recommendations, The Knowledge Engineering Review 35 (2020) e4. doi:10.1017/S0269888920000065.
[41] K. I. Kotis, G. A. Vouros, Human-centered ontology engineering: The HCOME methodology, Knowledge and Information Systems 10 (2006) 109–131. doi:10.1007/s10115-005-0227-4.
[42] M. Poveda-Villalón, A. Gómez-Pérez, M. C. Suárez-Figueroa, OOPS! (OntOlogy Pitfall Scanner!): An On-line Tool for Ontology Evaluation, International Journal on Semantic Web and Information Systems (IJSWIS) 10 (2014) 7–34. doi:10.4018/ijswis.2014040102.
[43] K. S. Kalyan, A survey of GPT-3 family large language models including ChatGPT and GPT-4, Natural Language Processing Journal 6 (2024) 100048. doi:10.1016/j.nlp.2023.100048.
[44] R. M. Bakker, D. L. Di Scala, From Text to Knowledge Graph: Comparing Relation Extraction Methods in a Practical Context, in: First International Workshop on Generative Neuro-Symbolic AI, co-located with ESWC 2024, Hersonissos, Crete, Greece, 2024.
[45] D. Garijo, O. Corcho, M. Poveda-Villalón, Foops!: An ontology pitfall scanner for the fair principles 2980 (2021). URL: http://ceur-ws.org/Vol-2980/paper321.pdf.
[46] A. Hemid, W. Shabbir, A. Khiat, C. Lange, C. Quix, S. Decker, OntoEditor: Real-Time Collaboration via Distributed Version Control for Ontology Development, in: A. Meroño Peñuela, A. Dimou, R. Troncy, O. Hartig, M. Acosta, M. Alam, H. Paulheim, P. Lisena (Eds.), The Semantic Web,

Springer Nature Switzerland, Cham, 2024, pp. 326–341. doi:10.1007/978-3-031-60626-7_18.
[47] A. Reiz, H. Dibowski, K. Sandkuhl, B. Lantow, Ontology Metrics as a Service (OMaaS), 2020, pp. 250–257. doi:10.5220/0010144002500257.
[48] A. Reiz, K. Sandkuhl, NEOntometrics: A Flexible and Scalable Software for Calculating Ontology Metrics, in: 18th International Conference on Semantic Systems, volume Vol-3235, CEUR Workshop Proceedings, Vienna, Austria, 2022. URL: ceur-ws.org/Vol-3235/paper16.pdf.
[49] D. Garijo, WIDOCO: A Wizard for Documenting Ontologies, in: C. d'Amato, M. Fernandez, V. Tamma, F. Lecue, P. Cudré-Mauroux, J. Sequeda, C. Lange, J. Heflin (Eds.), The Semantic Web – ISWC 2017, Springer International Publishing, Cham, 2017, pp. 94–102. doi:10.1007/978-3-319-68204-4_9.
[50] A. Alobaid, D. Garijo, M. Poveda-Villalón, I. Santana-Perez, A. Fernández-Izquierdo, O. Corcho, Automating ontology engineering support activities with OnToology, Journal of Web Semantics 57 (2019) 100472. doi:10.1016/j.websem.2018.09.003.
[51] D. Beßler, R. Porzel, M. Pomarlan, A. Vyas, S. Höffner, M. Beetz, R. Malaka, J. Bateman, Foundations of the Socio-Physical Model of Activities (SOMA) for Autonomous Robotic Agents, Formal Ontology in Information Systems 344 (2021) 159–174. doi:10.3233/FAIA210379, publisher: IOS Press.
[52] S. Borgo, R. Ferrario, A. Gangemi, N. Guarino, C. Masolo, D. Porello, E. M. Sanfilippo, L. Vieu, DOLCE: A descriptive ontology for linguistic and cognitive engineering, Applied Ontology 17 (2022) 45–69. doi:10.3233/AO-210259, publisher: IOS Press.
[53] LangChain, Rdflib, n.d. URL: https://python.langchain.com/docs/integrations/graphs/rdflib_sparql/, software documentation, version 0.3.
[54] C. Erlingsson, P. Brysiewicz, A hands-on guide to doing content analysis, African Journal of Emergency Medicine 7 (2017) 93–99. doi:10.1016/j.afjem.2017.08.001.
[55] C. Seljemo, S. Wiig, O. Røise, L. A. Ellis, J. Braithwaite, E. Ree, How Norwegian homecare managers tackled COVID-19 and displayed resilience-in-action: Multiple perspectives of frontline-staff, Journal of Contingencies and Crisis Management 32 (2024) e12558. doi:10.1111/1468-5973.12558.

A. Appendix

A.1. Requirements for the design of the human-LLM collaboration process framework

To extract the requirements inductively and directly from the users' needs and values, we have used the information gathered through the interview process for the exploration of the problem. As exemplified by Johannesson and Perjons [31], the requirements for the outlined artifact can follow from the root causes of the problem that the artifact will try to address and, ultimately, solve. As illustrated with the Ishikawa diagram presented in Section 4.1.2 (Figure 1), the problem addressed within this research is the complexity in the ontology extension process, and the root causes fall within different categories. However, solving these problems is not enough if the concerns and the opportunities that the ontology engineers perceive are not considered. Thus, based on the three themes explored within the content analysis of the interviews, the requirements aim to:

1. **Address** the complexity of ontology extension (Theme 1 - Figure 1).
2. **Reduce** the concerns about the use of LLMs for Ontology Engineering (Theme 2 - Figure 2).
3. **Leverage** the opportunities for the use of LLMs for Ontology Engineering (Theme 3 - Figure 3).

Table 2 shows the complete list of high-level requirements elicited from the interviews with the 11 professionals in OE and LLMs.

Table 2
Functional and non-functional requirements elicited from the interviews for the design of the process framework prototype.

Functional requirements
1.1. Support the acquisition of domain data.
1.2. Complement the role of the domain expert in tasks related to the acquisition of data and technical conceptualization.
1.3. Support ontology matching/alignment techniques.
1.4. Define the roles and responsibilities of the stakeholders and the LLM for each task.
1.5. Support backward compatibility of ontology changes.
1.6. Include CQs throughout the whole ontology extension development process.
1.7. Support various data source formats.
1.8. Facilitate the exploration of the ontology to be extended.
1.9. Support less experienced ontology engineers.
1.10. Facilitate comprehension of domain-specific information.
1.11. Include OE-tailored tools.
1.12. Facilitate documentation of ontology extension.
1.13. Promote reuse of existing ontologies and standards.
1.14. Support formulation and formalization of CQs.
1.15. Provide evaluation method for ontology extension including domain experts.
1.16. Specify format of LLM's output for each task.
Non-functional requirements (structural)
2.1. Outline the sequence of tasks.
2.2. Provide hybrid approach combining LLMs with traditional OE techniques.
2.3. Provide flexibility depending on the project characteristics, application and/or use case.
Non-functional requirements (environmental)
3.1. Promote the inclusion of diverse perspectives (include additional stakeholders in the process).
3.2. Be user-friendly and intuitive for ontology engineers with varying levels of expertise.
3.3. Encourage expert-in-the-loop-approach.

The transcripts of the interviews were analyzed using content analysis, following Erlingsson and Brysiewicz [54] and Seljemo et al. [55]. For each interview transcript, we extracted meaning units, condensed meaning units, and grouped these into categories, for each theme. An example of the analysis is provided below in Figure 11. We used the condensed meaning units to elicit the requirements. The

Interview #	Role of interviewee	Theme: The complexity of ontology extension		
		Meaning units	Condensed meaning units	Category
1	Ontology Engineer, background in formal logic	When concepts to add are very abstract, there is no unique correct way of modeling the data	High abstraction level	Process
			No single correct answer	Process
		There is a lot of discussion	Long discussions	Stakeholders
		It's difficult to extract the correct information from the domain expert	Extracting information from domain experts	Stakeholders
		Maintenance of ontologies is manual, changes must be manually reflected everywhere (implementation, visualization, documentation)	Manual maintenance	Tools

Figure 11: Example drawn from the analytic process for Theme 1: "The complexity of ontology extension".

resulting design requirements were validated within a focus group session with ontology engineers. From all requirements, only 1 requirement was not fulfilled: Requirement 1.5 (”Support backward compatibility of ontology changes”). Although crucial, its implementation is highly complex and challenging in practice.

A.2. Additional results from the demonstration and evaluation of the process framework

Table 3 below shows the Competency Questions generated by the GPT Assistant (based on GPT-4o) using our human-LLM collaboration process framework (Figure 4). This is the output of the task T-2.1: Formulate Competency Question in the process.

Table 3
Competency Questions generated by the GPT Assistant using our human-LLM collaboration process framework (Output of task T-2.1: Formulate Competency Questions).

Competency Question	Included
1. What type of obstacle has the robot detected?	Yes
2. Where is the robot currently located?	Yes
3. What is the severity level of the detected obstacle?	Yes
4. At what time was the obstacle detected?	Yes
5. What decision has the robot made regarding the detected obstacle?	Yes
6. Why has the robot made the specific decision for obstacle avoidance?	Yes
7. Is the detected obstacle a human?	Yes
8. Is the detected obstacle a structural component of the greenhouse?	Yes
9. What is the new location of the robot after obstacle avoidance?	Yes
10. How much time did the robot take to avoid the obstacle?	Yes
11. Were there any deviations from the original task after obstacle avoidance?	Yes
12. What static obstacle has been detected that requires human intervention?	Yes
13. How many obstacles have been detected within a specified time-frame?	Yes
14. What are the possible actions the robot can take after detecting an obstacle?	Yes
15. What are the recommendations from the robot for human intervention in case of a static obstacle?	Yes
16. How does the robot communicate its status post-obstacle avoidance?	No
17. What unexpected movable obstacle has been detected?	Yes

As an input for this task, the business scenarios output provided in task 1.5 was homogenized by restructuring the information exchanged between the Robot and the Human Operator and replacing the actions of the Human Operator by *Receives the notification and monitors the robot's decision* for all the business scenarios to reflect that the communication is only one-way, from the autonomous system to the human operator. Suggestions of new concepts to add to the ontology given in the input of task 1.1 were also added to the prompt.

As shown in Figure 5-a, after formulating the CQs, these shall be reviewed with the domain experts and other stakeholders such as the knowledge worker and the knowledge user. In this case, we compared these 17 CQs to the SENS gold standard (Figure 7) and we decided that 16 of them are relevant to the use case.

To verify that the generated ontology extension can answer the generated CQs (after populating the ontology in task T-4.1: Populate ontology extension with instances) we propose task T-4.2: Verify all CQs can be answered according to test cases. This task has been proposed previously in OntoChat [18]. The results of this task are shown in Table 4. From the 16 CQs, 14 have been correctly identified as "covered by the generated ontology extension". For *CQ2: Where is the robot currently located?*, the LLM incorrectly states that this question cannot be answered, however the ontology extension does include the object property hasLocation that assigns a Location to a Robot. For *CQ12: What static obstacle has been detected that requires human intervention?*, the LLM states that the question can be derived from the sub-class StaticObstacleAlert and the object property hasRecommendedAction, but this might only be possible if the data property hasExplanation contains the appropriate explanation about human intervention.

Finally, in Table 5 we provide a comparison summary between the ontology extension generated manually (SENS Gold Standard) and the ontology extension generated using our human-LLM collaboration framework, with the model GPT-4o (SENS-GPT).

Table 4
Verification of the generated ontology extension by the GPT Assistant using our human-LLM collaboration process framework (Output of task T-4.2: Verify all CQs can be answered according to test cases). Task based on the work of Zhang et al. [18].

Competency Question	GPT: Answered	GPT: Explanation	Correct / Incorrect
1. What type of obstacle has the robot detected?	Yes	The ontology includes the class Situation with properties like hasObstacleType and hasDetectedSituation which indicate the type of the detected obstacle.	Correct
2. Where is the robot currently located?	No	The ontology does not specify a property for the current location of a robot.	Incorrect
3. What is the severity level of the detected obstacle?	Yes	The ontology includes a Situation class with property hasSeverityLevel which can indicate the severity of a detected obstacle.	Correct
4. At what time was the obstacle detected?	Yes	The ontology includes a Situation class with property hasDetectedTime which specifies the detection time of an obstacle.	Correct
5. What decision has the robot made regarding the detected obstacle?	Yes	The ontology includes properties hasOutcomeAction, hasRecommendedAction within Situation and linked to Robot through actions.	Correct
6. Why has the robot made the specific decision for obstacle avoidance?	Yes	The ontology includes the hasExplanation, hasOutcomeAction, hasRecommendedAction within the Situation class.	Correct
7. Is the detected obstacle a human?	Yes	The property hasObstacleType in Situation can be used to specify if the obstacle is a Human.	Correct
8. Is the detected obstacle a structural component of the greenhouse?	Yes	The property hasObstacleType in Situation can be used to specify if the obstacle is a GreenhousePart.	Correct
9. What is the new location of the robot after obstacle avoidance?	No	The ontology does not include properties for specifying the new location after obstacle avoidance.	Correct
10. How much time did the robot take to avoid the obstacle?	Yes	The Situation class includes hasDetectedTime and hasOutcomeAction which further have time-related properties for calculating duration.	Correct
11. Were there any deviations from the original task after obstacle avoidance?	Yes	The hasTaskDeviation property within Action or OutcomeAction can be used to indicate deviations.	Correct
12. What static obstacle has been detected that requires human intervention?	Yes	The ontology includes StaticObstacleAlert subclass of Situation and can be linked to hasObstacleType and hasRecommendedAction.	Incorrect
13. How many obstacles have been detected within a specified time-frame?	Yes	The hasDetectedTime property within Situation can be used to filter based on time frames.	Correct
14. What are the possible actions the robot can take after detecting an obstacle?	Yes	The hasRecommendedAction property within Situation can be used for this purpose.	Correct
15. What are the recommendations from the robot for human intervention in case of a static obstacle?	Yes	The hasRecommendedAction property within StaticObstacleAlert can be used to specify recommendations.	Correct
16. What unexpected movable obstacle has been detected?	Yes	The ontology includes a UnexpectedMovableObstacle subclass of Situation and properties like hasObstacleType for unexpected movable obstacles.	Correct

Table 5
Overview of the main differences between the gold standard extension and the extension generated using the process framework with GPT-4o (SENS-GPT).

	SENS Gold Standard	**SENS-GPT**
Situations	Taxonomy with 7 sub-classes including different situations depending on the position of the human when detected as an obstacle	Taxonomy with 5 sub-classes (position of human not considered)
Obstacles	Extended from DUL ontology, including self-moving objects (such as humans) and movable objects (tools found in the greenhouse), and using the FixedObject already existing in the CGO	Obstacle class created, with three sub-classes: GreenhousePart (existing in the CGO), Human, and Object. Object has several sub-classes representing common tools found in the greenhouse
Severity level of situations	Object property of an unexpected situation. Severity is a class with instances (Alarm, Info, Warning) that depend on the characteristics of the object that have an impact in the robot's operation and the safety of the human	Data property of Situation. The data type is a string that can be "High", "Medium", or "Low" depending on the obstacle found and its impact on the operation of the robot and the safety of the human (reasoned by GPT)
Possible recognitions	Additional class to differentiate recognitions from reasoned situations in the SENS dashboard	Not included
Decision made by the robot when detecting an obstacle	Made outside the ontology, in the SENS dashboard using the data provided by the ontology model	Explicitly modeled within the ontology extension using the object properties hasAction and hasRecommendedAction and data properties such as hasExplanation and hasTaskDeviation
Reuse of existing ontologies	2 ontologies reused	Only 1 ontology identified for reuse
CQs	Some CQs were formulated during development but not documented	16 CQs generated and formalized, and used for verification of the ontology extension
Annotations (comments and labels)	Some annotations are missing	All annotations are included, but some are too generic
Glossary of terms	Not included	Basic glossary of terms automatically created from use case description