OntoChat Assistant for User Story Generation in Ontology Engineering

YIHANG ZHAO, King's College London, United Kingdom
ANELIA KURTEVA, University of Birmingham and King's College London, United Kingdom
ALBERT MEROÑO PEÑUELA, King's College London, United Kingdom
ELENA SIMPERL, King's College London, United Kingdom

An ontology is a formal, explicit specification of a shared conceptualisation, which can be combined with problem-solving methods and reasoning functionality to develop high-quality technology and application systems efficiently. Ontology engineering (OE) typically involves extensive manual effort to elicit intended use cases (user stories) from users for the target ontology-based systems. Recent studies have demonstrated the positive potential of LLM-based conversational agents in supporting user story generation in OE. However, we argue that we are not leveraging LLM to its fullest potential by not supporting users in formulating effective prompts. To address this, we identify the prompt guidance users need during user story generation workflows by conducting a formative study (N = 10) using participatory prompting. We demonstrate its usefulness through the design and development of the OntoChat LLM-based system for OE, as well as a user evaluation with knowledge engineers (N = 24). To our knowledge, this is the first work to design and validate a prompt guidance framework that helps users leverage LLM to its fullest potential to generate effective requirements for ontology development. This advances how we interact with LLM for requirements elicitation.

CCS Concepts: • **Human-centered computing** → **Human computer interaction (HCI)**.

Additional Key Words and Phrases: Ontology Engineering, Large Language Models, Interface Design

ACM Reference Format:
Yihang Zhao, Anelia Kurteva, Albert Meroño Peñuela, and Elena Simperl. 2018. OntoChat Assistant for User Story Generation in Ontology Engineering. *J. ACM* 37, 4, Article 111 (August 2018), 25 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Ontology, originally a philosophical term, refers to the study of being [30]. Several decades ago, ontologies were introduced into information and communication technologies (ICT) as a method for formally and explicitly representing the kinds of things (e.g., individuals, classes, attributes, interactions) that can be described within a system [27]. They can provide reusable declarative knowledge that can be integrated with problem-solving methods and reasoning functionality to build high-quality technology and application systems in an economical manner [66]. For example, they can support data and process integration, information retrieval, and information extraction [36].

Authors' Contact Information: Yihang Zhao, yihang.zhao@kcl.ac.uk, King's College London, London, United Kingdom; Anelia Kurteva, a.kurteva@bham.ac.uk, University of Birmingham, Birmingham and King's College London, London, United Kingdom; Albert Meroño Peñuela, albert.merono@kcl.ac.uk, King's College London, London, United Kingdom; Elena Simperl, elena.simperl@kcl.ac.uk, King's College London, London, United Kingdom.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM 1557-735X/2018/8-ART111
https://doi.org/XXXXXXX.XXXXXXX

Building and maintaining ontologies in an agile and collaborative manner has become a central paradigm of modern OE to ensure both usefulness and economic feasibility [66]. With the support of fully-fledged environments such as wikis and similar communication and collaboration platforms, open and geographically distributed groups of contributors or communities of practice can interact rapidly and iteratively to build a common understanding of a domain of interest [66, 74]. The knowledge developed through this collaborative process can then be structured for OE in terms of concepts, attributes, relationships, constraints, and more [66, 74].

An agile and collaborative OE process typically involves the elicitation of requirements imposed by the target ontology-based system from users [6, 66, 74]. Since ontologies are rarely used in isolation but rather as part of broader systems, it is not only essential that these requirements specify the domain knowledge to be formalised (as is done in most OE methodologies), but also at least as important to specify the intended use cases [36] that will be supported by the formalisation of that knowledge, for example, a collection of possible sequences of interactions between a system and its users that relate to a particular goal [6, 36]. To make use cases realistic, a possible approach is to collect them based on user stories provided directly by users [6, 13, 57].

User story 1: persona - Linka, computer scientist

1. Persona

Name: Linka, **Age**: 34, **Occupation**: Researcher in computer science, **Skills**: Expert in network data analysis and semantic web, **Interests**: Discovering hidden relationships in multi-modal music data.

2. Goal

Description: Linka aims to retrieve, integrate, and analyse multi-modal music datasets, enabling automated discovery of musical relationships. **Keywords**: music data, multi-modal data, automated discovery, large-scale integration.

3. Scenario

Before: Linka manually collects music data from multiple sources, struggling with entity matching (e.g., different spellings of artist names). The lack of standardization leads to errors and inefficiencies, making large-scale analysis impractical. **During**: She uses an ontology-based system to automate data retrieval, integration, and enrichment. It aligns data, resolves inconsistencies, and structures information for seamless analysis. **After**: Automated integration enables large-scale analysis, allowing Linka to explore genre classification, influence networks, and recommendation models with improved accuracy and efficiency.

(a) Ontology user story in the music domain

User story 2: persona - Sarah, political analyst

1. Persona

Name: Sarah, **Age**: 38, **Occupation**: Political Analyst, **Skills**: Data analysis, political research, journalism, **Interests**: Analysing political trends, elections, and disinformation narratives

2. Goal

Description: Sarah aims to analyse political candidates and governance by tracking trends, comparing election outcomes, and identifying disinformation patterns. **Keywords**: political analysis, elections, disinformation, insights

3. Scenario

Before: Sarah manually collects election data from multiple sources, struggling to track candidate performance and detect disinformation. The scattered data limits her ability to analyse trends and political narratives effectively. **During**: She uses an ontology-based system to integrate structured election data, visualize candidate relationships, and detect patterns in disinformation campaigns. **After**: Sarah generates in-depth, data-driven political reports, identifying hidden trends and relationships. The automated integration reduces manual effort, allowing her to focus on analysis and improving the accuracy of her insights.

(b) Ontology user story in the public sector domain

Fig. 1. Examples of standard ontology user stories

A user story can be formulated in different ways, ranging from one to three sentences up to three brief paragraphs, each focusing on one concrete part of the domain knowledge and one specific use case, to avoid excessive interpretation by knowledge engineers [6]. Among these formats, the IDEA framework [13], which extends XD [6], provides one of the most structured templates, introducing a user story with three parts [13, 87, 88] (see Fig. 1 for examples): (1) Persona—a profile describing a typical user details of the ontology-based system, such as name, age, occupation, skills, and interests; (2) User goal—a statement of what this user wants to achieve using the ontology-based system; and (3) Scenario—a description of the motivations, processes, and expected benefits through which the ontology-based system is applied or interacted with to help this user accomplish their goals.

By following this template, well-written user stories can represent different groups of users with common interests, frustrations, or desires. This helps build real empathy for these groups [6, 51] and to establish agreement on the design intent, scope, and criteria for success and completion, at least for a given time frame or a particular release of an ontology [6, 36]. Additionally, they will be used to transform into Competency Questions (CQs), which are single-sentence natural language representations of structured queries that the ontology should answer, and serve as guides for

ontology modelling and testing [6, 13, 87]. Finally, they can serve as a means to support informed judgments about whether an ontology is fit for reuse by enabling an easy comparison of the original goals and usage scenarios with the needs of a new project [36].

To elicit stories from users in large OE projects, manual synchronous elicitation methods such as workshops are often employed (e.g., in the ACCIO project [51]). However, it is challenging to accommodate everyone's availability for sufficiently long sessions, often resulting in workshops that are too brief to explore complex user requirements thoroughly. To address this, follow-up interviews [51] can be conducted after each session, but this approach demands significant time and effort from knowledge engineers who must interview many users. To reduce scheduling and time constraints, manual asynchronous methods such as collaborative spreadsheets can be used (e.g., in the Polifonia project [13, 14]). However, these approaches often lack real-time guidance and maintenance from knowledge engineers, which can lead to conflicting or poorly formulated requirements being collected from users.

In recent years, LLM-based systems have attracted increasing interest for supporting OE, as they can capture semantic relationships and contextual nuances within a domain and generate domain-relevant content for various OE tasks [2, 52, 53]. For example, Zhang et al. [87] demonstrated the positive potential of LLM-based conversational agents in supporting user story generation in OE. However, they also found that users, particularly those unfamiliar with prompting strategies, often struggle to devise effective prompts to elicit useful outputs, which is a common challenge identified in many other studies as well [38, 40, 86]. This highlights the importance of providing prompting support that helps users devise effective prompts [38, 40]. Therefore, we hypothesise that an LLM-based system can facilitate user story elicitation by offering prompt guidance to support effective interactions between users and the LLM at different interaction stages where users commonly encounter prompting trial and error. To validate this hypothesis, we formulate the following research questions (RQs).

- **RQ1:** What prompt guidance do users expect at each interaction stage of LLM-based ontology user story elicitation?
- **RQ2:** How can prompt guidance be integrated into an LLM-based system to support users at each interaction stage?
- **RQ3:** How useful is prompt guidance in supporting users during the ontology requirements elicitation workflow?

To address RQ1, we implement participatory prompting [19, 61, 88], which combines contextual inquiry [58] and participatory design [71], with researchers mediating interactions between users and the LLM. We use this method with knowledge engineers (N = 10) to identify the prompt guidance needed at each interaction stage. For RQ2, we design and implement the conversational agent prototype OntoChat, covering the interface, system workflow, and technical implementation, to guide users in prompting effectively. For RQ3, we evaluate the usability, utility, and effectiveness of prompt guidance using think-aloud protocols and post-task questionnaires (Likert scale) with knowledge engineers (N = 24). This paper contributes to LLM-based systems supporting collaborative OE by providing:

(1) A formative study (N=10) to identify the prompt guidance [1] users expect during different interaction stages of story generation in OE.

[1] https://github.com/King-s-Knowledge-Graph-Lab/OntoChat/blob/main/assets/user_study/User_Needs_for_the_LLM_assisted_Task_Assisting_in_User_Story_Creation.md

(2) OntoChat [2], an LLM-based conversational agent that demonstrates this prompt guidance with a user interface and interaction techniques designed to facilitate story generation in OE.
(3) A user study (N=24) demonstrating the usefulness of the prompt guidance in supporting story generation in OE, as presented in Section 5.3.

The structure of this paper is as follows: Section 2 reviews related works. Section 3 presents a formative study to establish the design goals of OntoChat. Section 4 describes the development and demonstration of OntoChat. Section 5 reports the user evaluation results. Section 6 discusses key findings and the study's limitations. Finally, Section 7 concludes the study and outlines directions for future work.

2 Related Works

We review (1) the status of ontology engineering methodologies (OEMs) in Section 2.1, (2) methods and LLM-based supports for ontology requirements elicitation in Section 2.2. Then, considering that LLM-based support for users in ontology user story elicitation has hardly been explored in the OE community, we examine (3) more mature LLM-based story elicitation approaches in software engineering and the creativity domain in Section 2.3.

2.1 Ontology engineering methodologies

Throughout the years, researchers in the Semantic Web community have proposed many OEMs to provide clear and complete guidance for OE activities. Almost all macro-level OEMs [67, 74, 80] (which specify the activities to transform an informal domain representation into an ontology) agree on three core steps : (1) conducting a feasibility study; (2) knowledge acquiring (including domain analysis, conceptualization, and transformation into an ontology using a formal language such as OWL); and (3) maintaining and updating the ontology to address new requirements.

Among macro-level OEMs, the earliest are Waterfall OEMs [22] (e.g. METHONTOLOGY [21]), which define an ordered sequence of steps from need clarification to ontology release. In these approaches, collaboration is not strongly emphasised; domain experts often play a passive role while ontology engineers lead the process and control the interaction. Later, Lifecycle OEMs [46] (e.g., UPON [16]) treat ontologies as evolving products involving human operators, processes, and technologies, allowing the ontology to evolve by passing through each phase in its lifecycle multiple times, not necessarily in a fixed order, with some phases occurring in parallel. The most recent evolution in OEMs is the adoption of Agile methodologies [55, 72], which reduce the role of ontology engineers to the final step of formalisation with ontological languages and instead focus on domain experts and stakeholders as co-participants in the engineering process to create a shared conceptualisation of the domain. These methods are widely used when it is important to involve the stakeholder community in development activities and to support participants with limited technical expertise.

Our work focuses on agile and collaborative OEMs. One of the first agile OEMs being proposed is XP.K [39] in 2002, which adapts Extreme Programming (XP) values to knowledge-based systems by generalising communication to the community and encourages teams to build collaborative infrastructure and foster respect for all team members' attitudes, backgrounds, and languages. In 2003, EXPLODE [32] was introduced as an agile method that uses CQs to guide requirement extraction and system constraints, enabling iterative development through CQs testing, planning, implementation, and continuous integration with frequent small releases. RapidOWL [5], proposed in 2006 and inspired by XP.K, incorporates Wiki-based concepts to promote joint development

[2] https://huggingface.co/spaces/1hangzhao/OntoChat

and iterative refinement of generic knowledge bases. Unlike XP.K, which targets specific scenarios, RapidOWL focuses on generic bases and involves domain experts as part-time knowledge engineers. While these early agile OEMs introduced collaborative and iterative principles, they often produced large monolithic ontologies without support for reuse or modularisation, which later methodologies sought to improve for better maintainability.

AMOD [24], proposed in 2014, is among the first approaches to support the creation and merging of independent ontology modules into a larger, generic ontology, based on groups of CQs produced from real-life scenarios (expert stories) provided by domain experts. Later, SAMOD [54], proposed in 2016, follows a similar process but specifies an iterative testing step: a produced ontology module based on a scenario is tested to check if it can answer all the CQs for the target ontology. If it cannot, the scenario is modified by adding more complexity or by developing and merging additional scenarios to produce a new ontology module until all CQs are addressed, making the final module the target ontology. While these agile OEMs support modular development to increase reuse and maintenance, domain experts are only involved in providing initial scenarios (stories), leaving ontology engineers with the majority of the burden for the detailed modelling and formalisation stages [74].

UPONLite [15], proposed in 2016, gives non-ontology specialists such as users (e.g., business experts) a central role by allowing them to use familiar tools like spreadsheets and conceptual maps to produce the domain conceptualization and specification, which reduces the involvement of ontology engineers until the final step of formalization. LOT [56], introduced in 2022, provides detailed guidance for collaborative activities, including use case specification and evaluation, which is currently lacking in UPONLite. AgiSCOnt [74], proposed in 2023, integrates concise, flexible, and adaptive developing instructions (currently lacking in most OEMs such as UPONLite, LOT, and SAMOD) throughout the OE process to support tasks such as exploring ontologies, comparing versions, debugging, and testing, especially for novice ontologists. Overall, these agile OEMs aim to shift responsibility for ontology building toward a community of non-ontology users through a social, highly participative approach supported by easy-to-use methods and tools. However, such extensive collaboration increases the manual effort required from knowledge engineers, ontology engineers, and users, highlighting the need for (semi-)automated OEMs in the era of LLM. More details on the role of these LLM-based OEMs in supporting requirements elicitation are provided in Section 2.2.

2.2 Ontology requirements elicitation

For ontology requirements elicitation, including domain knowledge, use cases, and more, the knowledge engineer is responsible for gathering requirements from users such as domain experts, contributors, stakeholders, or even themselves if they have relevant expertise or dual roles, or from available text corpora such as basic concepts and relationships within the domain.

To gather knowledge or use cases (stories) from users, unstructured interviews and workshops are helpful in the early stages when knowledge engineers lack familiarity with the domain. These interviews help engineers build empathy with users and gain an initial understanding before using structured techniques [10]. Once engineers gain some domain understanding (if it is through use cases, they need to extract key concepts), they can then use methods like card sorting [63], triad analysis [60], and twenty questions [35] to uncover how experts identify shared and unique characteristics among these domain concepts. Additionally, twenty questions [35] further determine the heuristics experts use in their problem-solving process. However, these manual elicitation techniques pose challenges, including scheduling and time constraints, where LLM-based (semi) automated systems can help. Existing work [87] used expert surveys and reported positive feedback on the potential of LLM-based conversational workflows to support ontology requirements

elicitation. However, the specific prompting support needed by users unfamiliar with prompting strategies for effective interaction during this process has not been systematically examined.

To gather knowledge from available text corpora, (semi-) automated frameworks are often used. These methods fall into two groups [4]: *classification-based NLP methods* [70] and *LLM-based approaches*. Classification-based (or clustering-based) NLP methods, such as named-entity recognition, information extraction, and analysis of lexical and syntactic features, have been used to help in three basic tasks of extracting knowledge from text: term extraction [33], synonym detection [31], and relationship extraction [43]. They utilise corpora, such as textbooks, journal articles, and other knowledge resources, to identify relevant terms and relationships. These terms and relationships are then combined automatically or with expert review [17, 31]. While these approaches can capture domain-specific knowledge, they often remain limited to specific domains due to their reliance on domain-specific patterns and expertise [17]. LLM-based approaches use large pre-trained models to capture broader semantic relationships and generate required content. For example, AutOnto [4], proposed in 2024, follows several key steps described in NLP-based tasks above (extracts relevant information from corpora, discovers information patterns), and also enriches extracted terms semantically to provide more context and clarity. Besides these, LLM-based approaches have also been used for other requirements elicitation tasks like generating and retrofitting CQs [1, 3, 9, 59]. They differ in the resources used for prompts. For example, AgOCQs [3] and NeOn-GPT [20] generate CQs from domain texts and controlled templates, with AgOCQs filtering outputs through semantic grouping. However, both face challenges in generalising to diverse or sparsely documented domains. When ontologies lack well-documented CQs, RevOnt [9] and RETROFIT-CQs [1] generate CQs from the ontology itself, improving documentation and reusability. The effectiveness of these methods still depends on the quality and structure of the underlying ontology. Despite the development of LLM-based frameworks for generating CQs from existing natural language text, the generation of use cases (stories) has received limited exploration.

2.3 User story elicitation

Ontologies are rarely used in isolation, but rather as part of an application or software system [6]. To determine what the ontology-based system or the ontology itself needs to provide, in what context, and for what purpose, use cases are essential [36]. They help define the scope of work, clarify criteria for success and completion, and support decisions about ontology reuse [6, 36]. Without clear scoping, ontology projects can lose focus and lack clear termination criteria [36]. One possible way to ensure use cases are realistic is to base them on user stories collected directly from users [6, 13, 57]. Users should primarily be stakeholders, which at least includes knowledge engineers and domain experts, since domain knowledge is provided by them for the specific purpose of an ontology-based system [6, 36]. The story elicitation process is usually the responsibility of knowledge engineers, who use methods such as interviews, workshops, or collaborative spreadsheets to collect stories and create a conceptual model of the domain from these stories [66]. They then represent this conceptual model in a suitable knowledge representation language that ontology engineers can understand [66].

Considering that the current extensive manual elicitation process is resource-intensive and existing LLM-based (semi-)automation frameworks are limited in this area, we therefore summarise research on LLM-based user story elicitation in software engineering and the creativity domain, where such approaches have been more extensively explored. This aims to understand better how LLM-based systems can streamline the process of gathering ontology requirements from human experts. In software engineering, user stories are often collected through structured dialogues. For example, Tumenjargal et al. [75] and Nakata et al. [45] develop chatbots that guide users in articulating needs, clarifying inputs, and identifying missing functionality or preferences. These

systems recognise user intent, extract relevant entities, and organise this information into actionable formats. They undergo iterative refinement based on user feedback, particularly in handling domain-specific terminology. However, these chatbots rely heavily on predefined conversational flows, which limits their adaptability to unexpected inputs and restricts the collection of richer contextual details. In the creativity domain, user stories (narratives) are often gathered through dynamic dialogues, focusing on contextual depth and emotional insights [68]. Wei et al. [81] design chatbots to collect self-reports through natural conversations, striking a balance between structured data collection and an open dialogue format. Kim et al. [37] develop MindfulDiary, which supports psychiatric patients in journaling through dynamically generated prompts and reflective questions, ensuring users are guided without feeling overwhelmed. Seo et al. [62] introduce ChaCha, a chatbot that facilitates emotional storytelling in children by building rapport, assisting in emotion identification, and utilising visual aids such as emojis for accessibility. However, user responses from these chatbots are not entirely reliable, as conversational flow is difficult to control, making goal-oriented elicitation challenging. Additionally, these systems struggle to maintain engagement in longer or more complex conversations. Based on these insights, collecting ontology user stories from users requires an LLM-based conversational agent to be designed with predefined interaction stages to ensure a goal-oriented process, while providing flexible support at each stage to facilitate effective user engagement.

3 Formative study

We conducted a formative study to understand the prompt guidance that users expect at each stage of interaction with LLM-based ontology user story elicitation. Section 3.1 introduces the LLM-based system we used, Section 3.2 introduces our participants, Section 3.3 details the study methods and procedure, Section 3.4 summarises the findings, and Section 3.5 outlines the resulting design goals.

3.1 Choice of LLM-based system

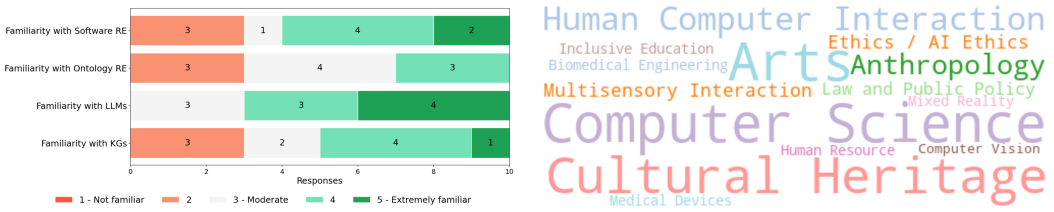
We chose the GPT-4o web interface [3], which is built upon a single LLM and includes UI elements and supporting modules, thereby replicating an advanced, realistic LLM-based system environment for user interaction. This choice was made because GPT-4o's web interface provides the ability to edit the original prompt based on responses, which is important in this user study for assessing the comparative quality of responses at each interaction stage. In addition, we compared GPT-4o's web interface with the Gemini 1.0 Ultra web interface by submitting example user queries at different stages of user story generation. The quality of responses and potential participant reactions to each system were evaluated, focusing on which would enable the most insightful interactions. This evaluation found that the GPT-4o web interface offers a more stable and responsive user experience and supports insightful interactions, both of which are key to ensuring the user study proceeds smoothly.

3.2 Participants

To understand the process of eliciting ontology user stories, we recruited 10 knowledge engineers specialising in OE, all of whom participated as volunteers. We selected them because they are expected to have experience in either requirements elicitation or translating user requirements into ontology modelling. This expertise enabled them to evaluate better which prompt guidance helps generate stories that are beneficial for ontology development, rather than producing interesting but non-essential stories.

[3] https://chatgpt.com/

To ensure diversity in participants' backgrounds and OE expertise, we conducted recruitment over a month through university mailing lists and MuseIT partners [4]. The recruitment covered 12 organisations, including research institutions, software development SMEs, and cultural and arts organisations from EU member countries, the United States, and the United Kingdom. However, the OE field is highly specialised, with fewer experts compared to broader areas such as machine learning or AI, making it challenging to find suitable participants. At the same time, we relied on voluntary participation to ensure genuine interest, which led to more meaningful data; however, this approach may have limited the number of suitable participants. In addition, the complex, multi-stage, and time-consuming nature of participatory prompting further limited recruitment. Finally, we recruited 10 participants: 7 are PhD researchers and 3 hold master's degrees; 6 work in academia and 4 work in industry.



(a) Familiarity with related technical backgrounds (b) Expertise domains of participants (excluding OE)

Fig. 2. Participant demographics distribution

We provide an overview of their demographic information in Fig. 2. We found that 3/10 knowledge engineers reported being only "slightly familiar" with ontology requirements engineering (ORE) or knowledge graphs (KGs). This can occur because some knowledge engineers focus mainly on knowledge representation (e.g., defining classes and properties) and have less involvement in requirements elicitation (e.g., gathering use cases), or vice versa. Including participants with diverse backgrounds can benefit the formative study by revealing different user needs for prompting support, which helps make the resulting prompt guidance framework more widely applicable. Ethical approval is granted by the King's College London Research Ethics Committee (24/01/2024, MRSP-23/24-41128), with all participants providing informed consent and no personal data collected.

3.3 Study process

We introduce the study method we use, participatory prompting [19, 61, 88], in Section 3.3.1. The whole user study script is available online [5]. Additionally, we provide an illustrative example [6], demonstrating the application of participatory prompting in the *user goal* generation subtask for one of our participants.

3.3.1 Participatory prompting. The participatory prompting method, first proposed by Sarkar et al. [61], is a user-centric research approach that combines principles of contextual inquiry [58] and participatory design [71]. In participatory prompting, researchers mediate participant interactions with a functional LLM-based system to identify the prompt guidance users need at each stage of interaction. The researcher-as-relay format [19] (illustrated in Fig. 3) is used: participants pose

[4] https://www.muse-it.eu/whoweare
[5] https://github.com/King-s-Knowledge-Graph-Lab/OntoChat/blob/main/assets/user_study/Study_Script_for_User_Story_Writing.md
[6] https://github.com/King-s-Knowledge-Graph-Lab/OntoChat/blob/main/assets/user_study/PID7_User_Goal_Description_Generation_Illustrative_Example.md

queries to the researcher, asking the LLM to generate a specific part of a story for a real OE project they have participated in. The researcher reformulates these queries using pre-identified prompting strategies and submits them to the LLM. Participants then review, reflect on, and build upon the model's responses to determine their next queries, guided by the researcher. This dialogue and participants' reflections on response satisfaction are analysed to identify effective prompt guidance needed during story generation. A key advantage of participatory prompting over low-fidelity prototyping [69] and Wizard-of-Oz methods [26, 42] is that it grounds studies in "actually existing AI" [7] capabilities rather than relying on simulations or speculative design probes. Compared to experiments with fully functional prototypes, it enables researchers to utilise off-the-shelf AI systems with minimal engineering effort and explore various use cases flexibly during the study, which may not be possible with a more constrained prototype.

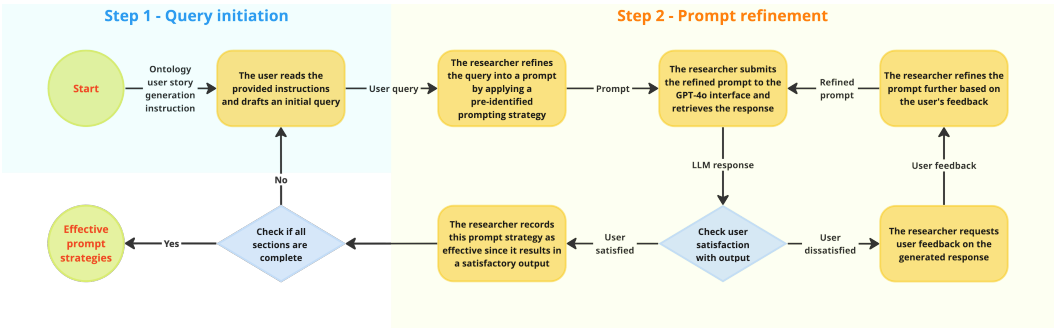


Fig. 3. Pipeline of participatory prompting

3.3.2 Step 1 - Query initiation. Participants are asked to use the GPT-4o web interface to generate a story for a real OE project they have participated in. They have been given an ontology user story template [7] to refer to, which includes three sections: *persona*, *user goal*, and *scenario*. This template is proposed by the IDEA framework [13], which extends XD [6] and has been widely adopted in OE [6, 13, 87, 88]. Its foundational framework is also well established in software engineering (SE), drawing on methodologies such as eXtreme Programming (XP) [65] and the experience factory [8].

Once participants have a query for LLM support in mind, they are asked to type it and send it to the researcher for refinement. Because participants are given a template to follow, we acknowledge that, as with any structured template, there is a risk that participants may feel limited in expressing their queries, or that the LLM may adhere too closely to the template, potentially restricting the evolution of story content. However, participatory prompting helps mitigate these issues by encouraging participants to reflect on the generated content and formulate follow-up queries based on their original intentions. This process guides the LLM to refine stories according to user needs, rather than rigidly following the initial template, which serves only as a reference.

3.3.3 Step 2 - Prompt refinement. The reason that participants send their queries to the researcher instead of directly to the LLM is due to a widely documented prompting challenge: users, especially those unfamiliar with prompting strategies, often struggle to devise effective prompts that elicit accurate and relevant outputs [38, 82, 86]. To address this limitation, participatory prompting involves the mediation of an expert researcher with knowledge and experience in prompt design,

[7] https://github.com/King-s-Knowledge-Graph-Lab/OntoChat/blob/main/assets/user_study/Ontology_User_Story_Template.png

who helps participants refine their initial query to an effective prompt by using pre-identified prompting strategies [8].

These pre-identified prompting strategies were independently developed by two researchers for the GPT-4o web interface over two weeks, using a range of real OE projects as story generation scenarios drawn from their own professional experience and followed widely recognized strategies from online tutorials, blogs, and research papers [40, 61]. At the end of this period, they met to discuss and formalise a list of effective prompting strategies and their application in different scenarios likely to arise during this study. These strategies incorporate elements such as system instructions, personas, constraints, tone, context, reasoning steps, few-shot examples, and response formatting as needed. However, even with a carefully designed strategy list, we found that many adjustments still had to be made ad hoc and in situ during practice.

The effective prompt, formulated by the researcher using the user query and pre-identified prompt template, is then submitted to GPT-4o's web interface, which generates a response. The researcher then engages the participant with reflection-oriented elicitation questions [5] to help assess whether the generated response meets their intentions. Based on the participant's feedback, the researcher iteratively adjusts the prompt and elicits new responses from GPT-4o until the participant is satisfied with the outcome. Once an effective prompting strategy that leads to a satisfactory response is identified, it is combined with other effective strategies (found in similar queries from other sessions) to create a reusable prompt template for future users. This process minimises repeated prompting trial and error, and supports more efficient interactions with the LLM. After addressing one user's query, the researcher guides the participant to continue generating the story until all parts are completed.

3.3.4 Data analysis. During each participatory prompting session, we collected demographic data (see Fig. 2) and recorded the screen sessions (user queries, prompting trials and errors, and LLM-generated responses) and the voice (participants' reflections and researcher guidance). We transcribed recordings from all 10 participants and anonymised them using PIDs. The qualitative analysis began with open coding, assigning 95 distinct codes to different prompt strategies validated as effective through a line-by-line review of the prompting trials, errors, and participants' reflections. This was followed by axial coding to group related codes into 43 broader categories (each category representing a user query), and thematic analysis to synthesise these categories into 8 overarching themes (each representing an interaction stage in story generation) To ensure consistency, the sole coder revisited the initial coding after a two-week interval to assess reliability. Discrepancies, which appeared in approximately 7% of the codes, were resolved through iterative revisions, resulting in a consistent coding scheme.

3.4 Findings

To address the challenge that users, especially those unfamiliar with prompting strategies, face in devising effective prompts that elicit accurate and relevant LLM outputs [38, 82, 86], we need to answer *RQ1 - What prompt guidance do users expect at each interaction stage of LLM-based ontology user story elicitation?* The interaction stages are identified by grouping user queries, which mark points where users commonly encounter prompting trial and error and need prompt guidance (see Section 3.3.4). After we identified the interaction stages, we examined what prompt guidance an LLM-based system can replicate from the researcher's role during the story generation process to support users in effective prompting when the researcher is not present. The researcher's role involves refining participants' queries into prompts using pre-identified prompting strategies. Therefore, to enable

[8] https://github.com/King-s-Knowledge-Graph-Lab/OntoChat/blob/main/assets/user_study/Pre_identified_Prompts_for_Ontology_User_Story_Elicitation.md

an LLM-based system to offer prompting strategies for users to refine their queries, we created reusable prompt templates by combining the effective prompt strategies (response constraints) within each interaction stage and generalising them with placeholders for user customisation. See Fig. 4 for an example template, created for the third interaction stage, "Action," where users are asked to describe the steps required to build an ontology-based system for their objective. The complete list of templates for each interaction stage is available online [1]. These prompt templates are widely applicable and can be used by future users, who can efficiently interact with the LLM by editing the placeholders in each template. This helps minimise prompting trial and error.

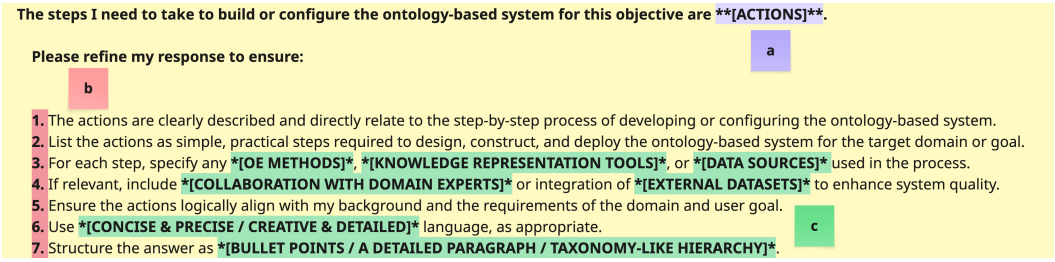


Fig. 4. Prompt template in OntoChat: (a) Mandatory placeholders, which users must fill in to provide basic information. (b) Optional placeholders, which allow users to add more detail; if left blank, the LLM will automatically fill them in the response to enhance the breadth and depth of the story. (c) Constraints, which include effective prompting strategies that guide the LLM to refine responses for clarity and relevance.

3.5 Design goals

To enable an LLM-based system to provide prompt templates [1] that support effective user prompting in the ontology user stories generation workflow, we identify four design goals for OntoChat to implement, building on the GPT-4o web interface used in our formative study.

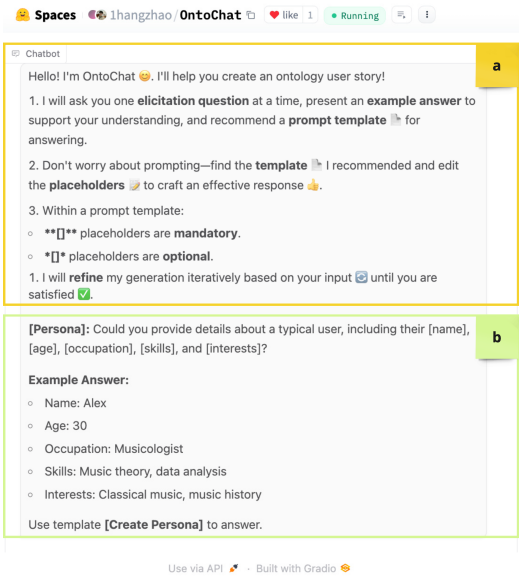
- **DG1:** OntoChat should propose elicitation questions, provide example answers, and recommend the most suitable prompt template [1] at each interaction stage to guide users in effective prompting during the story generation process.
- **DG2:** OntoChat should provide a template library to enable users to choose the suggested template.
- **DG3:** Each prompt template should be designed to support user customisation.
- **DG4:** OntoChat should iteratively ask users for feedback and refine the generated content based on their feedback until the user is satisfied.

4 OntoChat

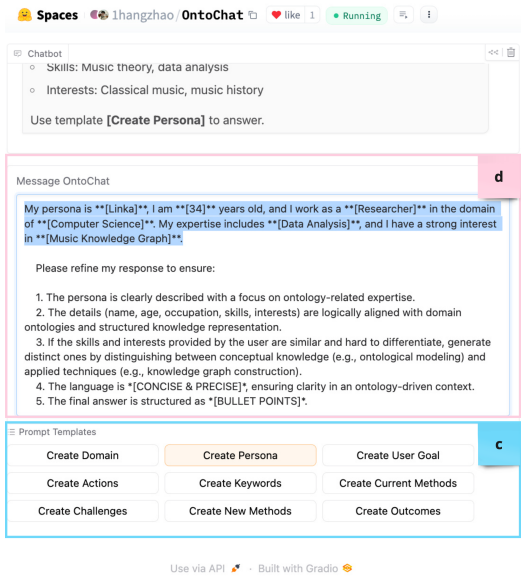
In this section, guided by the design goals, we present the OntoChat user interface in Section 4.1, demonstrate the story generation workflow in Section 4.2, and describe the technical implementation in Section 4.3.

4.1 Interface design

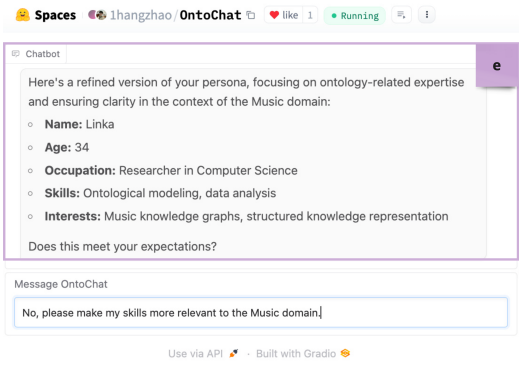
4.1.1 Conversational interface. We outline the conversational interface elements of OntoChat in Fig. 5, which consists of six components designed to guide users in effectively prompting during the ontology user story generation process.



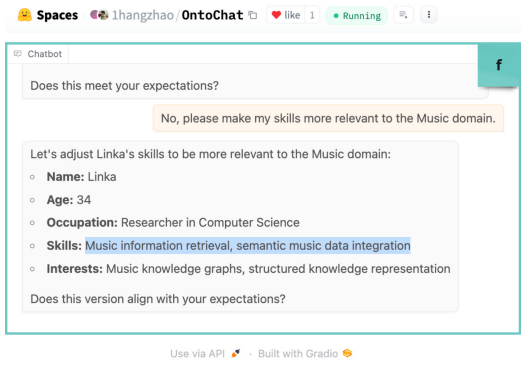
(a) Introductory (DG1). Shows the introductory message about the elicitation process and how to use prompt templates for first-time users. **(b) Elicitation question (DG1).** Begins the elicitation process by asking a real-time elicitation question, providing an example answer, and recommending a template.



(c) Prompt template library (DG2). Displays the templates for users to view and select; the chosen template is automatically populated into the user input box. **(d) User input box (DG3).** Allows users to write prompts from scratch or edit the selected template to tailor their responses.



(e) Generated output (DG4). Shows OntoChat's response generated from user input and structured in a story template.



(f) Generation refinement (DG4). Prompts the user for feedback and displays refined stories after each round of feedback.

Fig. 5. OntoChat user interface

4.2 System workflow

The system workflow of OntoChat, as shown in Fig. 6, consists of two main processes: user story elicitation (Fig. 6a) and user story refinement (Fig. 6b), supporting story generation in OE.

(a) User story elicitation workflow: Shows the guidance OntoChat provides to users during the story elicitation process to help them prompt effectively based on corresponding elicitation questions.

(b) User story refinement workflow: Shows the iterative story refinement process, which will terminate only if all parts are approved by users, to ensure the final story meets their expectations.

Fig. 6. System workflow of OntoChat user story generation

4.3 Technical implementation

OntoChat is developed as a web-based system hosted on Hugging Face Spaces[2] and is available under the MIT license on GitHub[9], supporting deployment on a local server. Its frontend prototype uses Gradio, as it provides a seamless framework for integrating LLM-based conversational functions with an intuitive user interface. The backend runs on Python 3.11 (required by Gradio) and leverages OpenAI's GPT-4o [34] API (which outperformed other LLM models for ontology user story generation in our pilot study period, as detailed in Section 3.1). For the backend prompts, we make them publicly accessible online in a separate file [10]. These prompts include: (1) *Persona*, assigning OntoChat the role of a knowledge engineer; (2) *System instructions*, defining tasks for assisting in ontology user story generation; (3) *Context*, incorporating relevant conversation history to support task execution; and (4) *Few-shot examples*, providing standard responses that demonstrate expected outputs for each story generation stage.

5 Evaluation

To answer *RQ3 – How useful is prompt guidance in supporting users during the ontology requirements elicitation workflow?*, we use widely recognised and validated system usefulness metrics, usability and utility, from Nielsen Norman Group [47] to evaluate how prompt templates support effective prompting in the story generation workflow. Similar to some technology acceptance metrics such as Technology Acceptance Model (TAM) [12] or Unified Theory of Acceptance and Use of Technology (UTAUT) [78], we assess perceived ease of use, behavioural intention, and result satisfaction. However, unlike TAM or UTAUT, which focus on overall user attitude and external influences [25], our metrics provide detailed usability and utility evaluation for each system feature. This allows us to understand how well each feature meets user needs for effective prompting, addressing a limitation of TAM and UTAUT, which pay little attention to the system feature-level fitness for specific user tasks [25]. To be more specific, our metrics: (1) evaluate each system feature's ease of use, overall efficiency, and behavioural intention as measures of EQ1 – Usability; (2) evaluate the extent to which functionality meets user needs as EQ2 – Utility; and (3) assess result generation for well-formedness, relevance, and helpfulness for OE as EQ3 – Effectiveness. The evaluation questions are as follows.

- **EQ1 - Usability:** To what extent do users find OntoChat's prompt templates easy to use?
- **EQ2 - Utility:** To what extent do prompt templates meet users' needs during the story generation process?

[9] https://github.com/King-s-Knowledge-Graph-Lab/OntoChat/tree/main
[10] https://github.com/King-s-Knowledge-Graph-Lab/OntoChat/blob/main/assets/user_study/OntoChat_Backend_Prompts.md

- **EQ3 - Effectiveness:** To what extent does the use of prompt templates result in satisfactory stories being generated?

For EQ1, we assess: *EQ1.1 User understanding:* To what extent do users understand the purpose of (*EQ1.1.1*) the prompt template library, (*EQ1.1.2*) the placeholders within each template, and (*EQ1.1.3*) the embedded constraints within each template? *EQ1.2 Ease of locating:* To what extent can users easily (*EQ1.2.1*) locate appropriate templates for each interaction stage, and (*EQ1.2.2*) identify all placeholders that need editing? *EQ1.3 Ease of using:* To what extent is it intuitive for users to (*EQ1.3.1*) select a template by clicking on it and (*EQ1.3.2*) modify template by editing placeholders? *EQ1.4 Efficiency:* To what extent do prompt templates help users generate stories quickly? *EQ1.5 Inclination:* To what extent are users inclined to utilise the provided templates?

For EQ2, we assess: *EQ2.1 Template library:* To what extent does the library provide the necessary templates at each interaction stage? *EQ2.2 Template placeholders:* To what extent do placeholders support user needs for personalising responses? *EQ2.3 Template embedded constraints:* To what extent do embedded constraints guide OntoChat's responses to align with user requirements?

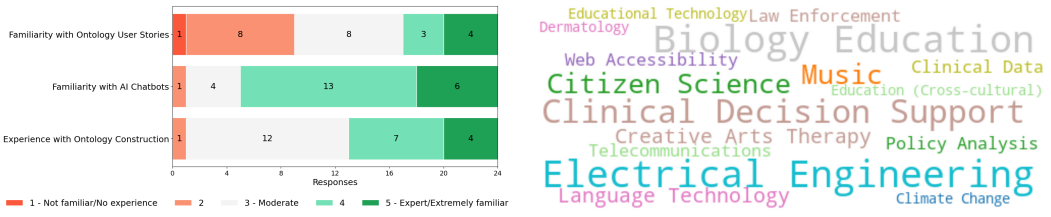
For EQ3, we assess: *EQ3.1 Relevance:* To what extent are the generated stories relevant to users' projects? *EQ3.2 Helpfulness:* To what extent are the generated stories helpful for the ontology construction process? *EQ3.3 Well-formedness:* To what extent do the generated stories have a complete structure?

For the remainder of this section, we introduce our participants in Section 5.1, describe our evaluation study using the think-aloud protocol in Section 5.2, and present the results of the user evaluation in Section 5.3.

5.1 Participants

To avoid bias from prior exposure, we recruited a new group of 24 knowledge engineers focused on OE to participate in the evaluation study, all as volunteers. This number is limited by the challenge of recruiting participants in a highly specialised domain, as discussed in Section 3.2. These participants are expected to have experience in either requirements elicitation or translating user requirements into ontology modelling, so they can assess whether the prompt guidance is useful for generating stories that benefit later ontology development.

To ensure participants with diverse backgrounds and OE expertise, we conducted recruitment over 2 months through the W3C Semantic Web public mailing list [11], an active forum for over 20 years where experts and newcomers from various backgrounds share knowledge and collaborate on advancing Semantic Web technologies. At the same time, to ensure the evaluation reflects a real OE process, participants were asked to use OntoChat to generate stories for their current or previous OE projects. An overview of their demographic information is provided in Fig. 7.



(a) Familiarity with related topics

(b) Participant project application domains in OE

Fig. 7. Participant demographics distribution

[11] https://lists.w3.org/Archives/Public/semantic-web/

We found that 9/24 knowledge engineers were at most "slightly familiar" with ontology user stories. This is likely because many focus mainly on CQs elicitation from corpora or knowledge representation (e.g., defining classes and properties), with little experience in story elicitation. Possible reasons include the projects they participated in often skipping use cases (stories) elicitation and moving directly to modelling due to time constraints, even though well-written use cases are recognised as critical for scoping ontology projects [36]. We accepted this variance, and during each session, we provided an introduction explaining ontology user stories and their uses, and clarified all participants' questions until none remained.

5.2 Study process

5.2.1 Think-aloud protocol and post-task questionnaire (Likert scale). We follow the pipeline in Fig. 8, beginning with the widely validated think-aloud protocols [77] for qualitative data collection. This includes Concurrent Think-Aloud (CTA) and Stimulated Retrospective Think-Aloud (RTA) [76]. In CTA, participants are introduced to background information [12], then asked to use OntoChat to generate an ontology user story for an OE project they have participated in, while verbalising their thoughts, emotions, and actions. This provides direct insights into their experience using prompt templates. However, CTA may disrupt natural workflows due to cognitive load. Therefore, CTA is commonly followed by RTA, conducted after the task using video recordings to support reflection. In RTA, participants are guided by open-ended questions to reflect on their challenges and the features they find most helpful when using prompt templates. However, think-aloud protocols alone may not capture users' overall perceptions in a quantifiable form suitable for comparison. Therefore, a Likert-scale questionnaire [13] is administered after the think-aloud sessions to let users quantify their perceptions of OntoChat's usefulness.

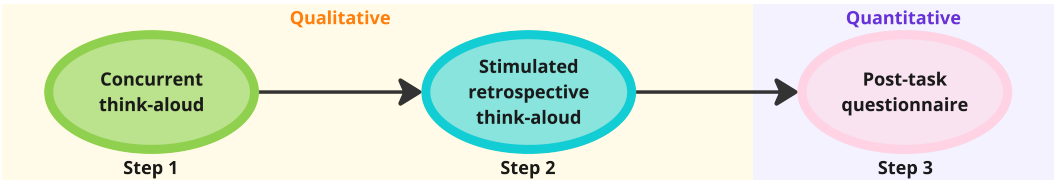


Fig. 8. Pipeline of user evaluation

Additionally, to capture and validate user and system behaviours during each interaction stage, observation checklists [14] are employed. These together help researchers triangulate qualitative and quantitative findings for a comprehensive evaluation of OntoChat.

5.2.2 Data analysis. The data analysis involves both qualitative and quantitative approaches. Screen data (including user queries, prompting trials and errors, and LLM-generated responses) and voice inputs (capturing participants' experience sharing and researcher guidance) from 24 participants are transcribed and anonymised using PID. The qualitative analysis includes open coding of participant experiences, with 230 distinct codes assigned through line-by-line review, followed by axial coding to group related codes into 53 broader categories (different challenges faced, points of satisfaction, and feature recommendations), and thematic analysis to synthesise

[12] https://github.com/King-s-Knowledge-Graph-Lab/OntoChat/blob/main/assets/user_study/Background_Information.md
[13] https://forms.gle/B83A93Jv9AAUffEA9
[14] https://github.com/King-s-Knowledge-Graph-Lab/OntoChat/blob/main/assets/user_study/Researcher_Observation_Checklist.md

these categories into 8 overarching themes (different interaction stages). To ensure consistency, the sole coder revisits the initial coding after a two-week interval to evaluate the reliability of the code application. During this review, discrepancies are identified in approximately 6% of the codes, particularly in feedback on the effectiveness of elicitation methods, which sometimes conflict within or across participants' responses. These discrepancies are resolved through iterative revisions, ensuring a consistent coding scheme is maintained. Quantitative data are analysed using statistical methods, with findings visualised using stacked bar charts, pie charts, and correlation heat maps. Data triangulation identifies convergence points between qualitative themes and quantitative results, enhancing the robustness of the analysis.

5.3 Results

To answer EQ1, EQ2, and EQ3 in Section 5, we present an analysis of post-task questionnaire responses in Fig. 9, and we then combine findings from researcher observations and think-aloud to describe participants' assessment of prompt templates' usefulness for effective prompting in story generation for each of EQs in following paragraphs.

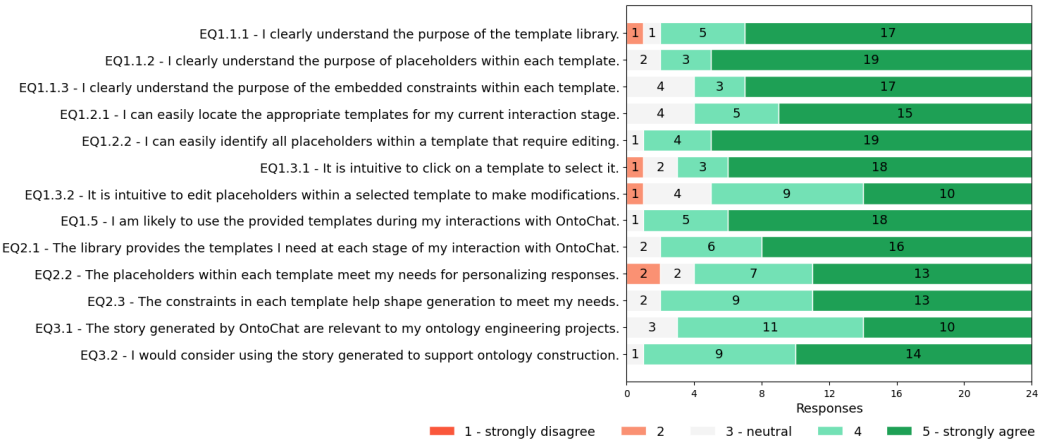


Fig. 9. User ratings on the usefulness of prompt templates

EQ1.1 evaluates the ease of understanding prompt templates. At least 22/24 participants rated at least 4 (easy to understand) for each of EQ1.1.1 (purpose of the prompt library), EQ1.1.2 (purpose of placeholders within templates), and EQ1.1.3 (purpose of embedded constraints within templates). However, one main confusion identified was about understanding the purpose of the prompt library. For example, PID 2 and PID 11 initially thought templates like "Create Persona" were navigation buttons and expected that clicking would open a new step. As PID 2 said, *"I think clicking the label will lead me to a new interface, but it doesn't work, so I am confused about what to do next."* After reading the introductory message and trying the system, they realised templates are for prompt crafting. They reflected that this confusion came from limited onboarding and a lack of visual cues, as they only received a static introductory message. Based on this, a potential solution is to add step-by-step interactive tutorials with examples and visual cues, as these can reduce users' cognitive load when learning a new system and enhance user sensemaking [23, 84].

EQ1.2 evaluates the ease of locating prompt templates and placeholders. At least 20/24 participants rated at least 4 (easy to locate) for both EQ1.2.1 (appropriate templates for their current interaction stage) and EQ1.2.2 (placeholders within templates that required personalisation). The

remaining participants did not encounter issues, but four suggested adding a "Click to use" button to automatically fill the input window with suggested templates without requiring a search of the library. This suggestion aligns with work on using quick-action buttons to reduce user effort and improve efficiency during interaction [28].

EQ1.3 evaluates the ease of using prompt templates. At least 19/24 participants rated at least 4 (easy to use) for both EQ1.3.1 (selecting a template by clicking on it) and EQ1.3.2 (modifying the template by editing placeholders). The remaining participants did not encounter major issues. Some minor issues were noted; for example, PID 11 did not perceive templates as clickable buttons due to the absence of arrows indicating clickability. However, this can be considered a niche individual preference, as all other participants used them without difficulty.

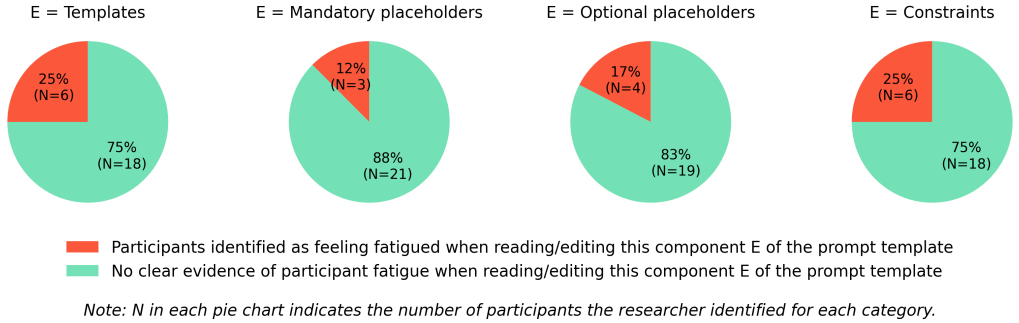


Fig. 10. The efficiency of using prompt templates to support users during story generation

EQ1.4 evaluates the efficiency of using prompt templates to help users generate stories. We analysed users' feedback on whether reading or editing prompt templates, placeholders, and constraints caused fatigue or boredom, which can reduce attention, motivation, and influence efficiency. Fig. 10 shows that 6/24 participants reported fatigue when reading templates and editing constraints. For example, PID 23 said, *"I feel tired to be responsible for editing so many constraints for so many templates to finish a single story!"* This may be because, although OntoChat uses the widely validated "chunking" principle [64] in HCI by breaking down story generation into smaller, manageable stages with template support to reduce cognitive load, too many similar steps (editing templates) make users feel they are repeating the same actions and must refocus often, which increases perceived effort and leads to fatigue. One possible solution is to reduce the "gulf of execution," which is the gap between what users want to do and how to do it in the system [50]. For example, auto-filling placeholders based on context could help reduce users' effort in editing templates.

EQ1.5 evaluates users' inclination to utilise the provided templates for interaction. At least 23/24 participants rated at least 4, indicating users are highly willing to use the templates. One possible reason is that prompt templates address a common mismatch challenge [19] between an LLM-based chatbot's actual capabilities and users' initial understanding in multi-step knowledge acquisition workflows. This challenge can lead to repetitive and frustrating interactions [28], where users try to figure out how to interact effectively through trial-and-error in text input boxes, often starting with generic prompts like *"Can you..."*. In contrast, prompt templates offer a low-interaction-cost approach [48, 49]. They present (1) the range of tasks the chatbot can handle through a template library overview, so users can quickly see what is possible; (2) pre-filled effective prompting strategies (constraints), so users do not need to worry about how to prompt effectively and can reduce trial-and-error for refinement; and (3) pre-filled constraints also clarify how responses are

generated, which helps users understand the process and increases their confidence in interacting with the chatbot.

EQ2 evaluates whether prompt templates meet users' needs. At least 20/24 participants rated at least 4 (satisfied) for each of the following: EQ2.1 (the library provides necessary templates at each interaction stage), EQ2.2 (the template provides necessary placeholders to help users personalise responses), and EQ2.3 (the template provides necessary constraints to guide OntoChat's responses). However, 5/24 participants found that additional prompting strategies (embedded constraints) are needed in the prompt template. They observed that OntoChat sometimes replaces technical jargon with semantically similar terms or introduces new processes without explanation, making it difficult to track and validate changes. Therefore, they expressed the need for prompting strategies to guide OntoChat to explicitly state what changes it made based on the last response and provide justification for those modifications. This aligns with research advocating for transparency in AI-driven OE [41].

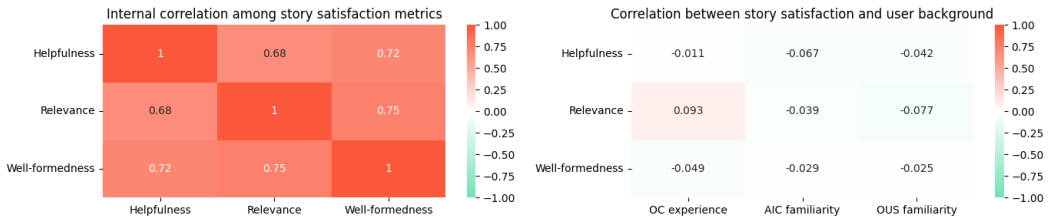


Fig. 11. Evaluating robustness of prompt template effectiveness

EQ3 evaluates the effectiveness of using prompt templates. At least 21/24 participants rated at least 4, indicating that the resulting stories are relevant to their project (EQ3.1) and the resulting stories are helpful for ontology construction (EQ3.2). For EQ3.2 (the well-formedness of generated stories), the researcher manually annotated each generated story's structure (1 for incomplete, 5 for fully complete) based on whether it contains all 8 artefacts of an ontology user story: persona, user goal, actions, keywords, current methods, challenges, new methods, and outcomes. Annotations were validated twice, with a one-week interval, and any discrepancies were resolved. We found that 12/24 generated stories were rated as 5 (fully complete), 10/24 as 4 (mostly complete), and 2/24 as 3 (partially complete). The main area of incompleteness was the "outcome" artefact, which often merely stated that the "user goal" was achieved, without describing the specific benefits provided by the ontology-based system. In addition, we assess the robustness of effectiveness. Fig. 11 presents the Spearman correlation coefficients among the story satisfaction metrics (helpfulness, relevance, well-formedness) and their correlations with user background factors. The consistently high internal correlations (all r in the range of [0.68, 0.75], all $p \leq 0.05$) among story satisfaction metrics indicate that OntoChat supports all aspects of story satisfaction together, not just one at the expense of others. The weak and non-significant correlations (all r in the range of [-0.077 0.093], all $p \geq 0.05$) with user background further indicate that prompt templates help users achieve satisfactory stories regardless of their prior experience with KG, user stories, or AI chatbots. Together, these findings confirm that the use of prompt templates is a robust (reliable and widely applicable) method for effective ontology story generation.

6 Discussion

In this section, we reflect on the participatory prompting methodology (Section 6.1), design goal fulfilment (Section 6.2), the usefulness of OntoChat for OE (Section 6.3), story elicitation methods (Section 6.4), quality metrics and datasets creation for LMs fine-tuning (Section 6.5).

6.1 Reflections on participatory prompting

Participatory prompting involves multiple cycles in which participants ask queries or provide feedback on previous outputs, and researchers convert these queries or feedback into prompts. These iterative cycles are time-intensive and sometimes discourage users from testing their initial ideas. Since every conversion requires researchers to spend time reviewing pre-identified templates and manually drafting prompts, all participants (5/5) in our pilot study tend to carefully consider each query or feedback before presenting it. This cautious approach results in some potential initial ideas remaining untested. Since we observed this during our pilot study, we mitigated it in our formative study by designing user prompts that actively encourage participants to provide feedback on the LLM's responses.

Additionally, as discussed in Section 3.3.3, the set of pre-identified prompting strategies was tested only in a limited set of real OE project scenarios in which the two researchers participated. Following the discussion, the researchers agreed that certain strategies were broadly effective across various domains. For example, asking for explanations to clarify logic ("Explain why you chose this approach"), adding more examples to make concepts clear ("Give another example of this concept"), breaking down tasks into smaller steps to reduce complexity ("List the steps needed to complete this task"), and using specific methods or frameworks to make the process more practical ("Apply the SMART criteria to define the goal"). However, strategies that require citations or utilise online documentation are not always effective. Many of the provided references are fabricated. Although we also tried prompts like "use highly cited articles" or "refer to [specific URL]," these approaches did not consistently improve the results. In the end, this study decided to abandon this type of strategy.

6.2 Reflection on design goal fulfilment

We used observation checklists [14] to log user and system activities during each session. Supported by questionnaire ratings and think-aloud feedback from Section 5.3, we find that all design goals are met. For DG1, the researcher observed that OntoChat consistently proposed the correct elicitation questions, example answers, and prompt templates for each interaction stage without errors, even when users moved back and forth to modify different parts of the story. For DG2, 22/24 participants found the library provides necessary templates at each interaction stage (EQ2.1), and 20/24 participants found it easy to locate the suggested templates for their current stage (EQ1.2.1). For DG3, 19/24 participants found it easy to customise templates by editing placeholders (EQ1.3.2), and 20/24 found the placeholders in each template met their needs for customisation. For DG4, the researcher observed that OntoChat consistently asked each user for feedback after generating outputs and did not proceed to the next step until user satisfaction was indicated. Although these existing design goals are met, additional design expectations have emerged. For example, 2/24 participants expected prompt templates to support additional languages. 5/24 participants expected OntoChat to enable retrieval-augmented generation using up-to-date domain-specific datasets, as LLM may have limited or outdated training data for some specialised domains.

6.3 Reflection on the usefulness of OntoChat for OE

OntoChat is designed to support the user story generation in OE. To understand its usefulness, it is essential first to consider the value of user stories in OE. As detailed in Section 1 and Section 2.3, user stories are essential for collecting realistic use cases from typical users, representing different user groups with common interests, frustrations, or needs [6, 51]. They help determine what the system should provide, in what context, and for what purpose. This information is important for (1) defining the scope of the ontology by clarifying requirements for project success and completion, as OE projects may otherwise continue for years and years without clear endpoints [6]; (2) providing a basis to extract CQs and their answers to guide ontology modelling and testing [6, 13, 87]; and (3) supporting decisions about ontology reuse by providing original usage scenarios and goals for comparison with new OE projects [6, 36].

Although many OE projects skip user story elicitation and move directly to modelling due to reasons such as time constraints and lack of guidance, this practice is not recommended by many studies [6, 36]. Here, we examine the causes of these challenges and how OntoChat can help address them. As detailed in Section 1, manual synchronous methods, such as workshops, are limited by participant availability and session length, making it challenging to explore complex requirements in depth [51]. Follow-up interviews can be helpful, but they require significant time and effort from knowledge engineers. Asynchronous methods such as collaborative spreadsheets [13, 14] reduce scheduling issues but often lack real-time guidance, leading to conflicting or poorly formulated requirements. OntoChat combines the flexibility of asynchronous methods, allowing users to generate requirements at any time and from any place, with the real-time guidance of synchronous methods. This helps users unfamiliar with prompting strategies to prompt effectively, enabling them to leverage LLM to its fullest potential and generate satisfactory user stories for ontology development. Based on results from Section 5.3, at least 18/24 participants found OntoChat easy to use and efficient, showing satisfactory usability. At least 20/24 participants reported that it met their prompting support needs during story generation, indicating satisfactory utility. At least 21/24 resulting stories were well-formed, relevant to users' current OE projects, and helpful for ontology construction, indicating satisfactory effectiveness. Therefore, with OntoChat, many OE projects can be encouraged to reconsider eliciting user stories, allowing the benefits of these stories to be incorporated and contributing positively to the overall OE process.

Additionally, current OE experts do not rely on any strict OE methodology [73, 79] for ontology developing, as the process is fragmented across many tools and workarounds, and there is no well-accepted framework or seamless toolchain for common OE tasks [73, 79]. For example, requirements elicitation often relies on generic tools, such as text editors and spreadsheets, and the elicited requirements are then often implemented in separate tools, such as Protégé [44]. This fragmented development process presents significant technical and resource challenges [87]. In contrast, OntoChat follows one of the best practices in OE, XD [6], and provides an LLM-based conversational framework with prompt guidance, with the ultimate aim to integrate key OE tasks into a streamlined workflow, supporting the process from story elicitation to CQ extraction from stories and ontology implementation based on CQs. This makes the OE process more accessible and effective for both experts and non-experts. Furthermore, this idea of using an LLM-based conversational framework with prompt guidance is not limited to benefiting OE methodologies that rely on user stories. It can be integrated with any human-involved process in LLM-based ontology development to support effective human-LLM interaction, resulting in more transparent, consistent, and satisfactory outcomes.

6.4 Reflections on story elicitation methods

The current requirements elicitation process of OntoChat begins with the system posing an elicitation question accompanied by an example answer to scaffold users in formulating their responses. While all participants (24/24) acknowledge that example answers help them understand how to respond, 14/24 also report that these examples steer their ideas toward the provided examples, potentially leading them to overlook better options. Drawing on creative thinking concepts [29, 83], this bias may arise because OntoChat offers limited support for the initial exploration of diverse ideas (divergent thinking). Relying on example answers can lead users to converge too soon on suboptimal ideas (convergent thinking), resulting in fixation. In this phenomenon, individuals overly focus on a single idea, hindering elicitation or creativity [11, 18, 85]. This aligns with prior studies [11, 85] that demonstrate experts often converge prematurely during creative processes, underscoring the human tendency to favour convergence and the inherent challenge of supporting divergent thinking. To mitigate fixation, OntoChat should incorporate mechanisms to foster both divergent and convergent thinking.

6.5 Reflections on quality metrics and datasets creation for LMs fine-tuning

Our study conducted extensive experiments to design quantitative criteria for evaluating the satisfaction of generated stories. We employed traditional NLP metrics such as "BLEU", "ROUGE", and "METEOR", alongside human expert-based metrics including "realism", "correlation", and "readability". However, we did not identify a single set of metrics that is universally applicable across different domains. For example, in the biology domain, user stories often contain many technical terms that cannot be easily simplified into more accessible language. Even when explanations are provided, they frequently employ other technical terms, resulting in low readability scores despite the stories being accurate and well-formed within their respective domains. Given the significant variation in story requirements across domains and the unique demands of different OE projects, the quality of stories in our study was primarily evaluated based on the subjective judgments of participants and researchers. Although we recruited participants with sufficient knowledge to assess the satisfaction of generation, to further improve the reliability of evaluations, one possible solution is to integrate an adaptive metric framework that dynamically adjusts the weight of evaluation criteria based on the target domain. This approach would allow for more consistent and meaningful quality assessments across diverse fields. Despite extensive research on fine-tuning language models for specialised tasks, which enables them to acquire domain-specific context and generate more relevant responses, the lack of domain-specific quality metrics for ontology user stories prevents the establishment of standardised annotations. This makes it challenging to develop well-annotated datasets needed for fine-tuning and benchmarking language models. Therefore, we advocate for the semantic web community to establish a set of agreed-upon quality metrics or annotation standards for ontology user stories, supporting the development and evaluation of language models in this domain.

7 Conclusions and Future Work

Our work proposes a prompt guidance framework to address the challenge that users, especially those unfamiliar with prompting strategies, often struggle to interact effectively with LLM and generate satisfactory user stories for ontology development. To understand what prompt guidance is needed, we employ participatory prompting, a user-centric method for identifying effective support users need at each stage of story generation. Based on these insights, we developed OntoChat, an LLM-based system that provides prompt template suggestions tailored to each interaction stage, allowing users to create effective prompts by simply customising the template. Our user

evaluation with knowledge engineers shows that prompt templates enable efficient prompt crafting and generate effective stories for ontology development. However, the current user evaluation is based on knowledge engineers, and including a wider range of users in the OE community is necessary to gain broader insights in future. To our knowledge, this is the first work to design and validate a prompt guidance framework that helps users leverage LLM to its fullest potential to generate satisfactory requirements for ontology development, which advances how we interact with LLM for requirements elicitation.

Acknowledgments

This study has received support from multiple sources, including co-funding by MuseIT under grant agreement No. 101061441 as part of the European Union's Horizon 2021-2027 research and innovation programme. SIEMENS AG and the Technical University of Munich, Institute for Advanced Study, Germany, have provided additional support. We also thank Prof. Jacopo de Berardinis, Dr. Neal Reeves, and Xi Hu for their valuable feedback during the writing process of this paper. Lastly, we extend our gratitude to all the evaluators who contributed valuable input on the system's usefulness and shared their user experiences.

References

[1] Reham Alharbi, Valentina Tamma, Floriana Grasso, and Terry Payne. 2024. An experiment in retrofitting competency questions for existing ontologies. In *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*. ACM, 1650–1658.

[2] Bradley P Allen, Lise Stork, and Paul Groth. 2023. Knowledge engineering using large language models. *arXiv preprint arXiv:2310.00637* (2023).

[3] Mary-Jane Antia and C Maria Keet. 2023. Automating the Generation of Competency Questions for Ontologies with AgOCQs. In *Iberoamerican Knowledge Graphs and Semantic Web Conference*. Springer, 213–227.

[4] Kiara Marnitt Ascencion Arevalo, Shruti Ambre, and Rene Dorsch. 2024. AutOnto: Towards A Semi-Automated Ontology Engineering Methodology. In *International Knowledge Graph and Semantic Web Conference*. Springer, 225–241.

[5] Sören Auer and Heinrich Herre. 2006. RapidOWL—An agile knowledge engineering methodology. In *International Andrei Ershov Memorial Conference on Perspectives of System Informatics*. Springer, 424–430.

[6] Eva Blomqvist, Karl Hammar, and Valentina Presutti. 2016. Engineering ontologies with patterns–the eXtreme design methodology. In *Ontology Engineering with Ontology Design Patterns*. IOS Press, 23–50.

[7] Erik Brynjolfsson. 2022. The turing trap: The promise & peril of human-like artificial intelligence. *Daedalus* 151, 2 (2022), 272–287.

[8] Victor R Basili1 Gianluigi Caldiera and H Dieter Rombach. 1994. The experience factory. *Encyclopedia of Software Eng* 1 (1994), 469–476.

[9] Fiorela Ciroku, Jacopo de Berardinis, Jongmo Kim, Albert Meroño-Peñuela, Valentina Presutti, and Elena Simperl. 2024. RevOnt: Reverse engineering of competency questions from knowledge graphs via language models. *Journal of Web Semantics* (2024), 100822.

[10] Nancy J Cooke. 1999. Knowledge elicitation. *Handbook of applied cognition* (1999), 479–509.

[11] Nigel Cross. 2004. Expertise in design: an overview. *Design studies* 25, 5 (2004), 427–441.

[12] Fred D Davis. 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly* (1989), 319–340.

[13] Jacopo de Berardinis, Valentina Anita Carriero, Nitisha Jain, Nicolas Lazzari, Albert Meroño-Peñuela, Andrea Poltronieri, and Valentina Presutti. 2023. The polifonia ontology network: Building a semantic backbone for musical heritage. In *International Semantic Web Conference*. Springer, 302–322.

[14] Jacopo de Berardinis, Valentina Anita Carriero, Albert Meroño-Peñuela, Andrea Poltronieri, and Valentina Presutti. 2023. The music meta ontology: a flexible semantic model for the interoperability of music metadata. *arXiv preprint arXiv:2311.03942* (2023).

[15] Antonio De Nicola and Michele Missikoff. 2016. A lightweight methodology for rapid ontology engineering. *Commun. ACM* 59, 3 (2016), 79–86.

[16] Antonio De Nicola, Michele Missikoff, and Roberto Navigli. 2005. A proposal for a unified process for ontology building: UPON. In *International conference on database and expert systems applications*. Springer, 655–664.

[17] Kristina Doing-Harris, Yarden Livnat, and Stephane Meystre. 2015. Automated concept and relationship extraction for the semi-automated ontology management (SEAM) system. *Journal of biomedical semantics* 6 (2015), 1–15.

[18] Steven P Dow, Alana Glassco, Jonathan Kass, Melissa Schwarz, Daniel L Schwartz, and Scott R Klemmer. 2010. Parallel prototyping leads to better design results, more divergence, and increased self-efficacy. *ACM Transactions on Computer-Human Interaction (TOCHI)* 17, 4 (2010), 1–24.

[19] Ian Drosos, Advait Sarkar, Xiaotong Xu, Carina Negreanu, Sean Rintel, and Lev Tankelevitch. 2024. "It's like a rubber duck that talks back": Understanding Generative AI-Assisted Data Analysis Workflows through a Participatory Prompting Study. In *Proceedings of the 3rd Annual Meeting of the Symposium on Human-Computer Interaction for Work*. 1–21.

[20] Nadeen Fathallah, Arunav Das, Stefano De Giorgis, Andrea Poltronieri, Peter Haase, and Liubov Kovriguina. 2024. Neon-GPT: a large language model-powered pipeline for ontology learning. In *European Semantic Web Conference*. Springer, 36–50.

[21] Mariano Fernández-López, Asunción Gómez-Pérez, and Natalia Juristo. 1997. Methontology: from ontological art towards ontological engineering. *Engineering Workshop on Ontological Engineering (AAAI97)* (1997).

[22] Alexander Garcia, Kieran O'Neill, Leyla Jael Garcia, Phillip Lord, Robert Stevens, Oscar Corcho, and Frank Gibson. 2010. Developing ontologies within decentralised settings. *Semantic e-Science* (2010), 99–139.

[23] Eun Go and S Shyam Sundar. 2019. Humanizing chatbots: The effects of visual, identity and conversational cues on humanness perceptions. *Computers in human behavior* 97 (2019), 304–316.

[24] Baby A Gobin. 2014. An agile and modular approach for developing ontologies. In *Technology development and platform enhancements for successful global e-government design*. IGI Global Scientific Publishing, 118–138.

[25] Dale L Goodhue. 2007. Comment on Benbasat and Barki's" Quo Vadis TAM" article. *Journal of the Association for Information Systems* 8, 4 (2007), 15.

[26] John D Gould, John Conti, and Todd Hovanyecz. 1983. Composing letters with a simulated listening typewriter. *Commun. ACM* 26, 4 (1983), 295–308.

[27] Nicola Guarino. 1998. *Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy*. Vol. 46. IOS press.

[28] Isabel Kathleen Fornell Haugeland, Asbjørn Følstad, Cameron Taylor, and Cato Alexander Bjørkli. 2022. Understanding the user experience of customer service chatbots: An experimental study of chatbot interaction design. *International Journal of Human-Computer Studies* 161 (2022), 102788.

[29] Devamardeep Hayatpur, Daniel Wigdor, and Haijun Xia. 2023. Crosscode: Multi-level visualization of program execution. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–13.

[30] Yuan He. 2024. *Language models for ontology engineering*. Ph. D. Dissertation. University of Oxford, Oxford, United Kingdom. https://doi.org/10.5287/ora-pd7q5y157

[31] Aron Henriksson, Hans Moen, Maria Skeppstedt, Vidas Daudaravičius, and Martin Duneld. 2014. Synonym extraction and abbreviation expansion with ensembles of semantic spaces. *Journal of biomedical semantics* 5 (2014), 1–25.

[32] MH Hristozova. 2003. EXPLODE: Extreme Programming for lightweight ontology development. *Master of Engineering thesis, Department of Computer Science and Software Engineering, The University of Melbourne* (2003).

[33] Shang-Hsien Hsieh, Hsien-Tang Lin, Nai-Wen Chi, Kuang-Wu Chou, and Ken-Yu Lin. 2011. Enabling the development of base domain ontology through extraction of knowledge from engineering domain handbooks. *Advanced Engineering Informatics* 25, 2 (2011), 288–296.

[34] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276* (2024).

[35] Elizabeth A Kemp. 1996. The role of the individual project in teaching knowledge acquisition. In *Proceedings 1996 International Conference Software Engineering: Education and Practice*. IEEE, 138–143.

[36] Elisa F Kendall and Deborah L McGuinness. 2019. *Ontology engineering*. Morgan & Claypool Publishers.

[37] Taewan Kim, Seolyeong Bae, Hyun Ah Kim, Su-woo Lee, Hwajung Hong, Chanmo Yang, and Young-Ho Kim. 2024. MindfulDiary: Harnessing Large Language Model to Support Psychiatric Patients' Journaling. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–20.

[38] Nils Knoth, Antonia Tolzin, Andreas Janson, and Jan Marco Leimeister. 2024. AI literacy and its implications for prompt engineering strategies. *Computers and Education: Artificial Intelligence* 6 (2024), 100225.

[39] Holger Knublauch. 2002. *An agile development methodology for knowledge-based systems including a Java framework for knowledge modeling and appropriate tool support*. Ph. D. Dissertation. Universität Ulm.

[40] Pawel Korzynski, Grzegorz Mazurek, Pamela Krzypkowska, and Artur Kurasinski. 2023. Artificial intelligence prompt engineering as a new digital competence: Analysis of generative AI technologies such as ChatGPT. *Entrepreneurial Business and Economics Review* 11, 3 (2023), 25–37.

[41] Elisavet Koutsiana, Johanna Walker, Michelle Nwachukwu, Albert Meroño-Peñuela, and Elena Simperl. 2024. Knowledge Prompting: How Knowledge Engineers Use Large Language Models. *arXiv preprint arXiv:2408.08878* (2024).

[42] Thomas K Landauer. 1986. Psychology as a mother of invention. *ACM SIGCHI Bulletin* 18, 4 (1986), 333–335.

[43] Kaihong Liu, WW Chapman, G Savova, CG Chute, N Sioutos, and Rebecca S Crowley. 2011. Effectiveness of lexico-syntactic pattern matching for ontology enrichment with clinical documents. *Methods of information in medicine* 50, 05 (2011), 397–407.

[44] Glaice Kelly Q Monfardini, Jordana S Salamon, and Monalessa P Barcellos. 2023. Use of competency questions in ontology engineering: A survey. In *International Conference on Conceptual Modeling*. Springer, 45–64.

[45] Takuya Nakata, Masahide Nakamura, Sinan Chen, and Sachio Saiki. 2024. Needs Companion: A Novel Approach to Continuous User Needs Sensing Using Virtual Agents and Large Language Models. *Sensors* 24, 21 (2024), 6814.

[46] Fabian M. Neuhaus, Steve Ray, and Ram D. Sriram. 2014. *Toward Ontology Evaluation Across the Life Cycle.* NIST Internal Report 8008. National Institute of Standards and Technology, Gaithersburg, MD. https://doi.org/10.6028/NIST.IR.8008

[47] Jakob Nielsen. 1993. *Usability Engineering.* Morgan Kaufmann.

[48] Nielsen Norman Group. 2024. New Users Need Support with Generative-AI Tools. Retrieved Jan. 25, 2025, from https://www.nngroup.com/articles/new-AI-users-onboarding/.

[49] Nielsen Norman Group. 2024. Prompt Controls in GenAI Chatbots: 4 Main Uses and Best Practices. Retrieved Jan. 25, 2025, from https://www.nngroup.com/articles/prompt-controls-genai/.

[50] Donald A Norman. 1986. Cognitive engineering. In *User centered system design*. CRC Press, 31–62.

[51] Femke Ongenae, Lizzy Bleumers, Nicky Sulmon, Mathijs Verstraete, Mieke Van Gils, An Jacobs, Saar De Zutter, Piet Verhoeve, and Ann Ackaert. 2011. Participatory design of a continuous care ontology-towards a user-driven ontology engineering methodology. *Knowledge Engineering and Ontology, Proceedings* (2011), 81–90.

[52] Jeff Z Pan, Simon Razniewski, Jan-Christoph Kalo, Sneha Singhania, Jiaoyan Chen, Stefan Dietze, Hajira Jabeen, Janna Omeliyanenko, Wen Zhang, Matteo Lissandrini, et al. 2023. Large language models and knowledge graphs: Opportunities and challenges. *arXiv preprint arXiv:2308.06374* (2023).

[53] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering* (2024).

[54] Silvio Peroni. 2016. A simplified agile methodology for ontology development. In *International Experiences and Directions Workshop on OWL*. Springer, 55–69.

[55] Silvio Peroni. 2017. A simplified agile methodology for ontology development. In *OWL: Experiences and Directions–Reasoner Evaluation: 13th International Workshop, OWLED 2016, and 5th International Workshop, ORE 2016, Bologna, Italy, November 20, 2016, Revised Selected Papers 13*. Springer, 55–69.

[56] María Poveda-Villalón, Alba Fernández-Izquierdo, Mariano Fernández-López, and Raúl García-Castro. 2022. LOT: An industrial oriented ontology engineering framework. *Engineering Applications of Artificial Intelligence* 111 (2022), 104755.

[57] Valentina Presutti, Eva Blomqvist, Enrico Daga, and Aldo Gangemi. 2011. Pattern-based ontology design. In *Ontology Engineering in a Networked World*. Springer, 35–64.

[58] Mary Elizabeth Raven and Alicia Flanders. 1996. Using contextual inquiry to learn about your audiences. *ACM SIGDOC Asterisk Journal of Computer Documentation* 20, 1 (1996), 1–13.

[59] Youssra Rebboud, Lionel Tailhardat, Pasquale Lisena, and Raphael Troncy. 2024. Can LLMs Generate Competency Questions?. In *ESWC 2024, Extended Semantic Web Conference*.

[60] Gery W Ryan, H Russell Bernard, et al. 2000. Data management and analysis methods. *Handbook of qualitative research* 2, 1 (2000), 769–802.

[61] Advait Sarkar, Ian Drosos, Rob Deline, Andrew D Gordon, Carina Negreanu, Sean Rintel, Jack Williams, and Benjamin Zorn. 2023. Participatory prompting: a user-centric research method for eliciting AI assistance opportunities in knowledge workflows. *arXiv preprint arXiv:2312.16633* (2023).

[62] Woosuk Seo, Chanmo Yang, and Young-Ho Kim. 2024. ChaCha: Leveraging Large Language Models to Prompt Children to Share Their Emotions about Personal Events. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–20.

[63] Nigel Shadbolt and A Mike Burton. 1989. The empirical study of knowledge elicitation techniques. *ACM SIGART Bulletin* 108 (1989), 15–18.

[64] Ben Shneiderman. 2010. *Designing the user interface: strategies for effective human-computer interaction.* Pearson Education India.

[65] James Shore and Shane Warden. 2021. *The art of agile development.* " O'Reilly Media, Inc.".

[66] Elena Simperl and Markus Luczak-Rösch. 2014. Collaborative ontology engineering: a survey. *The Knowledge Engineering Review* 29, 1 (2014), 101–131.

[67] Elena Paslaru Bontas Simperl and Christoph Tempich. 2006. Ontology engineering: A reality check. In *On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, GADA, and ODBASE 2006, Montpellier, France, October 29-November 3, 2006. Proceedings, Part I*. Springer, 836–854.

[68] Amira Skeggs, Ashish Mehta, Valerie Yap, Seray Ibrahim, Sean A. Munson, Aubrey Rhodes, James Gross, Predrag Klasnja, Amy Orben, and Petr Slovak. 2025. Micro-narratives: A Scalable Method for Eliciting Stories of People's Lived Experience. In *CHI*.

[69] Carolyn Snyder. 2003. *Paper prototyping: The fast and easy way to design and refine user interfaces*. Morgan Kaufmann.

[70] Riad Sonbol, Ghaida Rebdawi, and Nada Ghneim. 2022. The use of nlp-based text representation techniques to support requirement engineering tasks: A systematic mapping review. *Ieee Access* 10 (2022), 62811–62830.

[71] Clay Spinuzzi. 2005. The methodology of participatory design. *Technical communication* 52, 2 (2005), 163–174.

[72] Daniele Spoladore and Elena Pessot. 2021. Collaborative ontology engineering methodologies for the development of decision support systems: Case studies in the healthcare domain. *Electronics* 10, 9 (2021), 1060.

[73] Daniele Spoladore and Elena Pessot. 2022. An evaluation of agile ontology engineering methodologies for the digital transformation of companies. *Computers in Industry* 140 (2022), 103690.

[74] Daniele Spoladore, Elena Pessot, and Alberto Trombetta. 2023. A novel agile ontology engineering methodology for supporting organizations in collaborative ontology development. *Computers in Industry* 151 (2023), 103979.

[75] Altansukh Tumenjargal and Sergey Balan. 2024. Requirements Elicitation From User Feedback Using Real-Time Conversational AI. Bachelor's Thesis, University of Gothenburg, Gothenburg, Sweden. Retrieved from https://gupea.ub.gu.se/handle/2077/84482.

[76] Maaike Van Den Haak, Menno De Jong, and Peter Jan Schellens. 2003. Retrospective vs. concurrent think-aloud protocols: testing the usability of an online library catalogue. *Behaviour & information technology* 22, 5 (2003), 339–351.

[77] Maarten Van Someren, Yvonne F Barnard, and J Sandberg. 1994. The think aloud method: a practical approach to modelling cognitive. *London: AcademicPress* 11, 6 (1994).

[78] Viswanath Venkatesh, Michael G Morris, Gordon B Davis, and Fred D Davis. 2003. User acceptance of information technology: Toward a unified view. *MIS quarterly* (2003), 425–478.

[79] Markel Vigo, Samantha Bail, Caroline Jay, and Robert Stevens. 2014. Overcoming the pitfalls of ontology authoring: Strategies and implications for tool design. *International Journal of Human-Computer Studies* 72, 12 (2014), 835–845.

[80] Markel Vigo, Caroline Jay, and Robert Stevens. 2014. Design insights for the next wave ontology authoring tools. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1555–1558.

[81] Jing Wei, Sungdong Kim, Hyunhoon Jung, and Young-Ho Kim. 2024. Leveraging large language models to power chatbots for collecting user self-reported data. *Proceedings of the ACM on Human-Computer Interaction* 8, CSCW1 (2024), 1–35.

[82] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382* (2023).

[83] Robin H Willemsen, Isabelle C de Vink, Evelyn H Kroesbergen, and Ard W Lazonder. 2023. The role of creative thinking in children's scientific reasoning. *Thinking Skills and Creativity* 49 (2023), 101375.

[84] Su-Fang Yeh, Meng-Hsin Wu, Tze-Yu Chen, Yen-Chun Lin, XiJing Chang, You-Hsuan Chiang, and Yung-Ju Chang. 2022. How to guide task-oriented chatbot users, and when: A mixed-methods study of combinations of chatbot guidance types and timings. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–16.

[85] Robert J Youmans and Thomaz Arciszewski. 2014. Design fixation: Classifications and modern methods of prevention. *AI EDAM* 28, 2 (2014), 129–137.

[86] J Diego Zamfirescu-Pereira, Richmond Y Wong, Bjoern Hartmann, and Qian Yang. 2023. Why Johnny can't prompt: how non-AI experts try (and fail) to design LLM prompts. In *Proceedings of the 2023 CHI conference on human factors in computing systems*. 1–21.

[87] Bohui Zhang, Valentina Anita Carriero, Katrin Schreiberhuber, Stefani Tsaneva, Lucía Sánchez González, Jongmo Kim, and Jacopo de Berardinis. 2024. OntoChat: a framework for conversational ontology engineering using language models. In *European Semantic Web Conference*. Springer, 102–121.

[88] Yihang Zhao, Bohui Zhang, Xi Hu, Shuyin Ouyang, Jongmo Kim, Nitisha Jain, Jacopo de Berardinis, Albert Meroño-Peñuela, and Elena Simperl. 2024. Improving Ontology Requirements Engineering with OntoChat and Participatory Prompting. In *Proceedings of the AAAI Symposium Series*, Vol. 4. 253–257.