A Survey on Interaction Design with Large Language Models for Ontology Requirements Elicitation with Competency Questions

Yihang Zhao [a], Xi Hu [a], Timothy Neate [a], Albert Meroño Peñuela [a] and Elena Simperl [a]
[a] *Department of Informatics, King's College London, London, United Kingdom*
E-mails: yihang.zhao@kcl.ac.uk, xi.hu@kcl.ac.uk, timothy.neate@kcl.ac.uk, albert.merono@kcl.ac.uk, elena.simperl@kcl.ac.uk

Abstract. Competency questions (CQs) are essential in ontology engineering (OE), as they express an ontology's functional goals and serve as a foundation for its construction, evaluation, and reuse. Large language model (LLM)-based systems for CQs elicitation have recently attracted substantial attention. These systems can generate thousands of candidate CQs from domain experts or knowledge sources to help define the boundaries of a target application domain. However, current interaction paradigms fall short in supporting knowledge engineers auditing target domain boundaries to ensure that CQs are neither too few, risking critical omissions, nor too many, resulting in information overload. This gap reflects the absence of supports for processes that closely align with the divergent (lateral) and convergent (vertical) thinking observed in the arts and creativity domain. We therefore present a systematic literature review (N = 50) investigating interaction design patterns in LLM-based systems that support divergent–convergent thinking in the arts and creativity domain. We then map the identified patterns to the context of CQs elicitation and propose an interaction model that extends the existing OntoChat system, enabling knowledge engineers to navigate a set of candidate CQs on an interactive canvas, supporting exploration, evaluation, and informed decisions about what to add, discard, or revise in order to progressively define the scope of the target domain.

Keywords: Ontology engineering, Requirements elicitation, Large language models, Interface design, Interaction paradigms

1. Introduction

Competency questions (CQs) are essential in OE, as they express an ontology's functional goals and serve as a foundation for its construction, evaluation, and reuse [6]. With the advent of large language models (LLMs), a shift emerged in 2023 toward the (semi-)automation of OE activities, particularly CQs elicitation [4–7, 11, 35, 45, 132, 134]. During this step, several LLM-based systems have been developed to assist knowledge engineers in defining the boundaries of a target domain by identifying relevant CQs—either by functioning as conversational agents, such as OntoChat [132, 134], which interactively elicits ontology user stories from domain experts and subsequently converts them into CQs, or by serving as automated tools that generate CQs from diverse knowledge sources. For example, AgOCQs [11] and NeOn-GPT [45] derive CQs from domain-specific text corpora and controlled templates, while RevOnt [35] reverse-engineers and RETROFIT-CQs [4] retrofits CQs based on existing ontology vocabularies and structures.

One of the main challenges in the CQs elicitation step is determining whether an elicited CQ is relevant—namely, auditing the boundaries of the target domain lexicon to assess whether the terminology suggested by a CQ is appropriate [4–7, 41]. Auditing this boundary is not easy, since in practice, domain boundaries are fluid—meaning

1570-0844/$35.00 © 0 – IOS Press. All rights reserved.

that terminology from one domain often connects with terminology in another [41]. Knowledge engineers must therefore consider two key dimensions [39, 41]. Along the "horizontal dimension," they must consider the domain coverage of the sought ontology, which refers to the breadth of terminology that needs to be included. For example, when defining an ontology in the medical domain, they must consider how much of the technology domain—such as terminology related to electro-medical devices—should be included. Along the "vertical dimension," they must consider terminology granularity, which concerns both the terminological box (TBox) and the assertional box (ABox) [12]. However, existing LLM-based systems focus solely on generating CQs, without supporting knowledge engineers in navigating across a set of CQs candidates to explore, evaluate, and make informed decisions about what to add, discard, or revise in order to progressively define the scope of the target domain [41, 104, 134]. This gap reflects the absence of support for processes that closely align with the divergent (lateral) and convergent (vertical) thinking observed in the arts and creativity domain, where this cognitive process is commonly used to support designers in systematically exploring, evaluating, and refining a set of ideals to converge on a design goal [40, 53, 104, 122].

At the same time, considering that LLM-based systems have already been widely applied to support users' divergent–convergent thinking in the arts and creativity domain—we present a systematic literature review (N = 50) investigating the interaction design patterns [22, 51, 110], including interaction techniques and user interface (UI) designs, used in LLM-based systems that support this process. We then map the identified patterns to the context of CQs elicitation, with the goal of supporting knowledge engineers in seamlessly exploring, evaluating, and refining a set of CQs candidates to progressively define the boundaries of the target application domain. To achieve this goal, we formulate the following research questions:

- **RQ1:** What interaction techniques are required with LLM to facilitate divergent–convergent thinking?
- **RQ2:** What UI designs are required with LLM to implement these interaction techniques?
- **RQ3:** What supporting can these interaction design patterns provide for knowledge engineers during CQs elicitation?

To address RQ1 and RQ2, we conducted a systematic literature review (SLR) to identify papers on LLM-based systems designed to support divergent–convergent thinking in the arts and creativity domain. We analysed and synthesized recurring interaction techniques and UI designs used to support this cognitive process. For RQ3, we adapted the identified interaction design patterns to the cognitive processes observed in CQs elicitation and proposed an interaction model that extends the existing OntoChat system, demonstrating how these patterns can support knowledge engineers in auditing and refining ontology boundaries. Additionally, since many CQs elicitation workflows begin with the collection of user stories from domain experts—which are later transformed into CQs—we also includes cognitive process involved in user story elicitation to inform the interaction model design more comprehensively. Our contributions are as follows:

1. A taxonomy consisting of 13 categories and 21 subcategories, developed to analyse and organize interaction techniques (Section 4.1) and UI designs (Section 4.2) that support divergent–convergent thinking in LLM-based systems.
2. A publicly available codebook[1] developed from the proposed taxonomy, consisting of 50 annotated LLM-based systems, each labelled with the interaction techniques and UI designs they employ to support divergent–convergent thinking.
3. An interaction model (Section 4.3) developed based on the identified interaction techniques and UI designs, extending the existing OntoChat system to support knowledge engineers' divergent–convergent thinking during requirements (CQs) elicitation.

The proposed interaction model is designed to support both academic researchers and practitioners in OE and HCI in developing LLM-based systems by enhancing human–LLM interaction—particularly in scenarios where human input is essential for progressively expanding and refining a latent conceptual space to support sensemaking toward a defined target goal. The structure of this paper is as follows: Section 2 introduces background on ontology requirements elicitation and divergent–convergent thinking. Section 3 presents the methodology of our SLR and

[1] https://github.com/King-s-Knowledge-Graph-Lab/OntoChat/blob/main/assets/systematic_literature_review/Codebook.xlsx

mapping method. Section 4 describes the resulting codebook and details the proposed interaction model. Section 5 reflects on the mapping approach. Section 6 outlines the study's limitations. Finally, Section 7 concludes the paper and outlines directions for future work.

2. Background

2.1. Ontology requirements elicitation

Ontology requirements elicitation—or knowledge acquisition—refers to the process by which a knowledge engineer gathers knowledge from domain experts or existing knowledge sources (e.g., domain-specific text corpora or ontology vocabularies and structures) [86, 89, 90, 102]. These requirements suggest the concepts and relationships the target ontology should model, thereby defining its scope and serving as references for its design, evaluation, maintenance, and reuse [67].

To gather knowledge from domain experts—particularly when knowledge engineers are unfamiliar with the domain at early stages—unstructured interviews are often used to explore user stories, including expert personas, goals, and the gaps the target ontology aims to address, which are then used to construct CQs [37, 39, 132]. However, while knowledge engineer–facilitated interviews help ensure that the elicited user stories are well-structured and relevant to ontology construction, such interactions often fall short in stimulating domain experts' divergent thinking. This is because the conversation tends to be anchored in the knowledge engineer's existing mental model [84]—meaning the process primarily builds on what the knowledge engineer already knows and expects. When knowledge engineers possess only limited understanding of the domain, this anchoring may further restrict their ability to support domain experts in imagining broader user stories [84].

Once knowledge engineers acquire some initial understanding, they typically shift toward structured elicitation methods to transform user stories into a more distilled understanding of the domain—clarifying key concepts, their properties, and the distinctions between them, which collectively form the foundation for constructing CQs. Common structured elicitation approaches include card sorting, where domain experts group related terms to reveal conceptual hierarchies and category structures [97]; triad analysis, where experts are asked to identify which two of three concepts are most similar and why, revealing subtle distinctions [94]; and the twenty questions technique, where a guessing-game format surfaces implicit properties, constraints, or assumptions tied to domain concepts [66]. However, these techniques follow a predominantly convergent path, building on predefined knowledge in the form of user stories acquired from initial interviews to formalize CQs, yet leaving limited opportunity for conceptual expansion. Additionally, they are time-intensive and may be impractical for large-scale projects involving numerous domain experts.

In response to these challenges, recent LLM-based systems such as OntoChat [132, 134] emulate the role of a knowledge elicitor by gathering and refining user stories through prompt-driven conversational workflows, which are then automatically converted into CQs. This approach aims to stimulate user divergence through follow-up elicitation questions and a variety of suggested answers. However, preliminary user studies indicate that current conversational interfaces are often linear—focusing on quick refinement based on the user's initial input—rather than encouraging broader reflection or exploration of alternatives [103, 104, 134]. As a result, users may become fixated on the pathway shaped by their initial input [38, 44, 63, 104, 130], limiting opportunities to explore alternative user stories and ultimately leading to CQs that do not adequately reflect the intended ontology scope.

Another approach to ontology requirements elicitation is to extract knowledge from existing domain documentation. Traditionally, this involves manually authoring CQs based on established guidelines [52, 86], which rely on general knowledge of lexico-syntactic patterns used to express CQs. However, this process often results in inconsistencies due to variations in authoring styles and subjective interpretations by different knowledge engineers [6]. In response, several approaches have emerged to support the semi-automated authoring of CQs by guiding knowledge engineers through predefined templates, often implemented using Controlled Natural Language (CNL) [70]. Methods such as CLaRO [65] and others [10, 20, 92, 124] derive syntactic patterns from existing CQ datasets to create structured CQ templates. These templates are then filled with domain-specific content by mapping ontology elements—such as classes and properties—into appropriate placeholders, typically using Natural Language Processing

(NLP) techniques. However, these approaches still require extensive manual template design for domain adaptation, due to their shallow understanding of semantics [6]. To address these limitations, recent approaches have begun to leverage LLMs to generate CQs through carefully crafted prompts that incorporate various knowledge sources. For example, AgOCQs [11] and NeOn-GPT [45] generate CQs from domain-specific text corpora using controlled templates, with AgOCQs further filtering results using semantic grouping to ensure quality. Meanwhile, both RevOnt [35] and RETROFIT-CQs [4] generate CQs from existing ontology vocabularies—RevOnt by reverse-engineering and RETROFIT-CQs by retrofitting. However, these approaches still require manual effort to validate the generated CQs in terms of correctness, relevance, coverage, etc. Additionally, these approaches lack support for knowledge engineers in navigating across the set of generated CQs, in order to explore and refine them toward better scoping the boundaries of the target application domain—one of the ultimate goals of ontology requirements elicitation.

2.2. Divergent-convergent thinking

The support process currently lacking for knowledge engineers in scoping domain boundaries during ontology requirements elicitation closely parallels the divergent–convergent thinking observed in the arts and creativity domain. This cognitive process is often used to support designers in systematically exploring, evaluating, and refining a set of ideas to reach a design goal [40, 53, 104, 122].

Divergent (lateral) thinking [53, 122] refers to the process of generating multiple, unique ideas toward a design goal, whereas convergent (vertical) thinking [40] involves evaluating and synthesizing those ideas to arrive at a single creative outcome. Prior research shows that the interplay between the two is key to avoiding fixation [38, 44, 130]—a tendency to stick with initial ideas—and supports broader, more effective idea generation [103, 104]. It also boosts confidence in one's ability to make informed decisions in open-ended tasks. In our context, divergent thinking can be interpreted as the requirements exploration process, where knowledge engineers generate as many unique CQs as possible to explore the boundaries of the target domain. Conversely, convergent thinking refers to the requirements refinement process, where knowledge engineers evaluate and synthesize the generated CQs to arrive at an appropriate subset that effectively scopes the target application domain—serving as a foundation for ontology construction.

Divergent–convergent thinking relies on the structural support of a design space [17, 21, 120], which refers to a conceptual "space of ideas organized by dimensions," broadly understood as collections of ideas, designs, concepts, or solutions [55, 80]. One common way to represent a design space is as a Cartesian space, where variations are distributed along two defined dimensions. Each variation reflects specific values on those dimensions, and each point in the space represents a unique combination of values [17, 55, 76, 98]. This kind of Cartesian space representation of design space is widely familiar in brainstorming tools such as Miro [2], where participants arrange sticky notes on an open canvas to externalize and organize ideas. Although Miro does not enforce a fixed two-dimensional structure, teams often assign meaning to the horizontal and vertical axes (dimensions) and populate the resulting grid with ideas, examples, or concepts corresponding to each combination. For example, when using Miro to externalize a narrative design space, users might define one axis as *genre* (e.g., fantasy, mystery) and the other as *tone* (e.g., uplifting, dark), then place each narrative idea at the intersection of its corresponding genre and tone—such as a story labelled *fantasy* and *uplifting*. In our context, the externalized design space of candidate CQs can also be organized along two dimensions. For instance, in the medical diagnostics domain, the horizontal dimension may represent domain coverage (e.g., with dimension values such as *body systems*, *symptoms*, *diagnostic tools*, or *treatment plans*), while the vertical dimension may represent terminology granularity (e.g., with values indicating depth levels within the TBox or ABox). A point in this space represents a terminology, such as *lung*, suggested by a candidate CQ like *"What organs are parts of body systems?"*, within a specific subdomain (e.g., *body systems*) and at a specific level of TBox granularity (e.g., based on the path length from *lung* to *entity* if using WordNet [81] as a reference)

Since an externalized design space makes both dimensions and their values explicit, it enables dimensional reasoning [5, 17], meaning that designers can systematically reflect on these dimensions to evaluate different variations

[2] http://miro.com/

and identify a desired direction toward achieving their design goals. In our context, dimensional reasoning enables knowledge engineers to reflect on the domain coverage and granularity of terminology suggested by CQs by stepping back from individual candidates and considering the entire space of CQs at a higher conceptual level. This all-encompassing view [121] makes it easier for knowledge engineers to continuously compare CQs dimensions, and to decide where to diverge or converge in order to effectively audit the boundary of the target ontology.

2.3. Supporting Divergent-Convergent Thinking

To support divergent–convergent thinking with LLM-based systems, we investigate interaction design patterns [22, 51, 110] drawn from information visualization [99] and human–computer interaction (HCI) [104]. These patterns are particularly suitable for enabling users to externalize and manipulate abstract information through spatial representations, helping them uncover patterns, gaps, and relationships while reducing cognitive load.

Interaction design patterns include both interaction techniques and UI designs [22, 51, 110], and they vary depending on input modalities such as screen-mouse-keyboard, gestures, body gestures, and others [27]. In this work, we specifically focus on interaction design patterns based on the screen–mouse–keyboard modality. Under this modality, interaction techniques refer to specific user actions—such as clicking, dragging, hovering, and scrolling—that enable task completion through interaction with the system. These actions shape how user input is interpreted and how system feedback is delivered [62, 129]. These techniques are supported by UI designs [43, 85], which provide the visual and functional components that allow users to express intentions and receive feedback. Common examples include buttons, sliders, menus, and input fields.

A central reference point for applying these interaction design patterns to support divergent-convergent thinking is Shneiderman's "Visual Information-Seeking Mantra" [99]—*"Overview first, zoom and filter, then details-on-demand"*. This principle outlines a sequence of interaction design patterns that guide users through broad exploration of the data space (divergence), followed by a gradual narrowing of focus to examine specific items in detail (convergence), thereby facilitating sensemaking in complex information spaces.

Many existing LLM-based systems [1, 8, 33, 54, 103, 104, 125] follow Shneiderman's framework to support divergent–convergent thinking in pursuit of their target art creation. We therefore summarize the commonly observed interaction techniques and their corresponding UI designs in these systems, organized according to the sequential structure of Shneiderman's framework. Here, we start with "Overview" [15, 48, 95], which is supported by interaction techniques such as panning, zooming out, and spatial navigation, along with UI designs like multidimensional spaces, timelines, networks, hierarchical trees, etc., which provide users with a high-level understanding of the information space and help identify areas of interest. Then, we move to "Zoom and filter" [3, 18, 64, 123], which involves interaction techniques such as zooming in, querying, brushing, and faceted selection, implemented through UI deigns like sliders, checkboxes, scroll controls, search bars, etc., enabling users to focus on relevant subsets, reduce visual clutter, and uncover finer patterns without losing context. Finally, we arrive at "Details-on-demand" [99], which is achieved through interaction techniques such as mouse hover, click-to-expand, or drill-down actions, realized via UI designs like tooltips, expandable panels, pop-up windows, clickable data items, etc., that reveal additional information when required. Together, these interaction techniques and UI designs reduce cognitive load by externalizing structure, preserve context across interaction levels, and support sensemaking through smooth transitions from broad exploration to focused analysis. However, a holistic understanding of the interaction techniques and UI designs currently used in LLM-based systems to support divergent–convergent thinking remains lacking, and their mapping to ontology requirements elicitation has yet to be explored.

3. Methodology

To address Research Questions 1 and 2, we conducted a systematic literature review (SLR) following the PRISMA 2020 guidelines [88]. The full SLR process is illustrated in Figure 1. Section 3.1 details the identification stage, including query formulation and venue filtering; Section 3.2 outlines the screening stage; and Section 3.3 covers the snowballing stage, including both forward and backward reference searches. Section 3.4 describes the development of the codebook used to analyse the included papers. To address Research Question 3, we conducted a design

thinking workshop structured by the Double Diamond model [14, 42], as described in Section 3.5, which guided the mapping process and informed the development of the proposed interaction model.

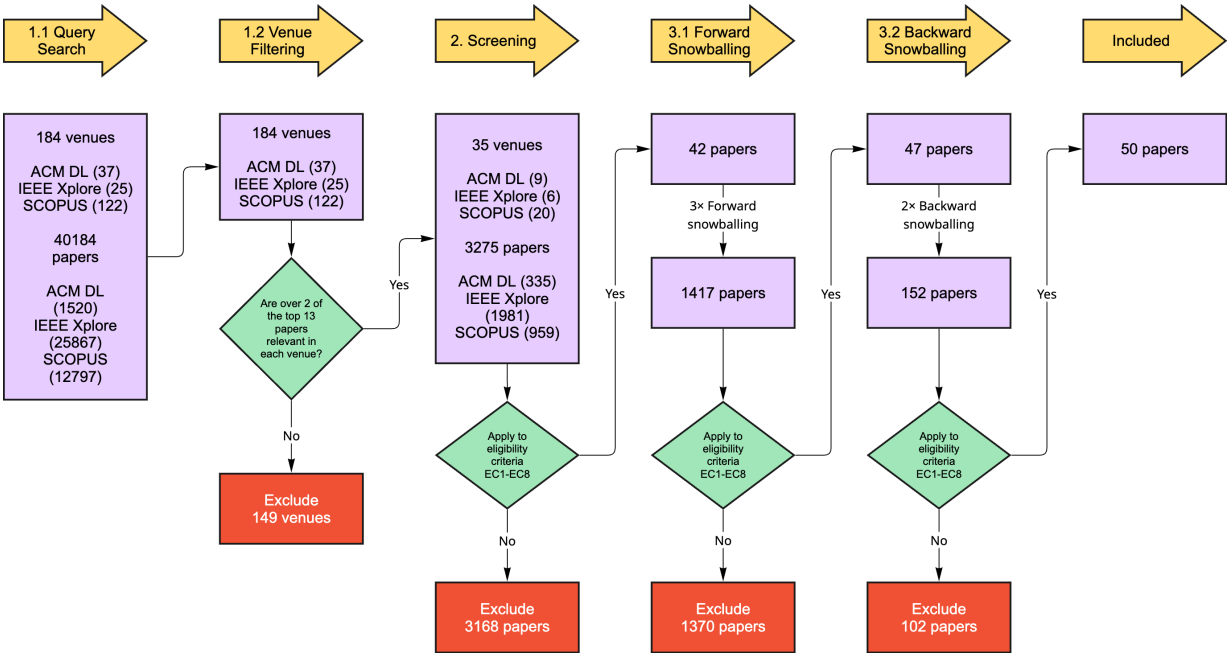


Fig. 1. The SLR pipeline follows the PRISMA 2020 guidelines [88]

3.1. Identification

To identify relevant literature on the design of LLM-based systems that integrate interaction techniques and UI designs to support designers' divergent-convergent thinking in art creation, we conducted a structured search across three major digital databases: the ACM Digital Library[3], IEEE Xplore[4], and Scopus[5]. These databases were selected due to their broad indexing of well-known publication venues in generative AI (GenAI), HCI, interactive system design, and creative computing, including *CHI*, *UIST*, *IUI*, *C&C*, *ICCC*, and *AIIDE*.

In formulating the search query, we intentionally did not constrain the search terms to those explicitly associated with LLMs. This decision was motivated by the modular integration of LLMs in creative systems, where they often support specific stages within broader multimodal or multi-step pipelines that also delegate other computational tasks—such as image generation, music synthesis, or style transfer—to specialized models. We therefore adopted broader umbrella terms such as `AI` and `machine learning` to ensure comprehensive coverage of relevant literature. The final search query was constructed as follows:

```
("artificial intelligence" OR "AI" OR "machine learning" OR
"large language model" OR "large language models" OR LLM*)
        AND generat* AND (art* OR creati*)
```

After some preliminary search trials, we found that the use of umbrella terms resulted in a large number of irrelevant papers (e.g., on classifiers, regressors, or clustering models). We therefore selected the publication date

[3]https://dl.acm.org/
[4]https://ieeexplore.ieee.org/Xplore/home.jsp
[5]https://www.scopus.com/

range starting from 2017, as it coincides with the release of the Transformer architecture [112], a key milestone in the evolution of generative models. Additionally, we observed that the term `art*` often produced false positives in the main body of papers (e.g., in phrases like `state-of-the-art`). To improve precision, we limited our query to search only within the *title*, *abstract*, and *keywords* fields. Following these settings, the search yielded a total of 40,184 non-duplicate records across the three databases. To make the dataset more manageable, we retained only conference and journal publications and excluded venues that were predominantly unrelated to our topic. Relevance was assessed by excluding venues in which fewer than 10% of their top 12 retrieved papers were deemed relevant. The cut-off of 12 was selected to ensure a consistent threshold, as it represents the smallest number of papers retrieved from any included venue. Ultimately, we retained 3,275 papers for the screening phase.

3.2. Screening

To effectively screen the papers, we defined eight eligibility criteria (EC), as detailed in Table 1. During the initial screening based on `EC-1` to `EC-4`, we did not restrict the paper types to full papers, as short papers, workshops, symposiums, and demos often present inspiring interaction design as well. Therefore, our primary focus was to assess whether the paper introduces inspiring interaction techniques and UI designs within an LLM-based system, and whether these are evaluated for their effectiveness in supporting the creative process—specifically, divergent–convergent thinking in the context of artistic expression or creative design.

EC-#	Description
`EC-1`	The full text of the paper must be available in English.
`EC-2`	The paper must not have been retracted.
`EC-3`	The paper must be peer-reviewed.
`EC-4`	The paper must not be a secondary study (e.g., a survey).
`EC-5`	The paper must present an LLM-based system that supports creative processes in artistic expression or creative design.
`EC-6`	The LLM-based system must introduce inspiring interaction techniques that contribute to the creative process.
`EC-7`	The LLM-based system must incorporate inspiring UI designs that facilitate the interaction techniques.
`EC-8`	The interaction design introduced in the paper must be evaluated to show their effectiveness in supporting the creative process.

Table 1
Eligibility criteria (EC) used in the screening process

The first two authors independently screened a random sample of 20% ($N = 655$) of the papers to calibrate their application of the eligibility criteria based on the title and abstract. We calculated the inter-rater reliability (IRR) using Fleiss' kappa [47], obtaining a score of 0.82, which indicates an "almost perfect" level of agreement (0.81-0.99) between the two authors. The first two authors also resolved disagreements in inclusion decisions through discussion until consensus was reached, primarily concerning the judgment of whether the artifacts generated by the LLM-based system could be considered an instance of artistic expression or creative design (`EC-5`). Following the screening process, a large number of papers were excluded by `EC-6`, as they focused on fully automated generation pipelines or model-centric optimization without clearly defined interaction design. After applying all criteria, a total of $N = 42$ papers were included.

3.3. Snowballing

To ensure better paper coverage, we conducted three iterations of forward snowballing, followed by two additional iterations of backward snowballing, all performed in 2025 to confirm that no further relevant papers were overlooked. For forward snowballing, we used Google Scholar[6] to retrieve the "Cited by" list for each included paper. For backward snowballing, we examined the reference lists of included papers to identify titles that appeared

[6]https://scholar.google.com/

relevant. In total, we identified $N = 8$ additional papers after removing duplicates and applying the screening criteria. When combined with the 42 papers retained from the initial screening phase, this brought the final dataset to $N = 50$ papers, with publication years ranging from 2017 to 2025.

3.4. Analysis

To analyse the 50 papers and address our research questions, we developed a codebook using a hybrid approach that combined both top-down [50] and bottom-up strategies [26, 109]. This methodological choice is supported by prior work in qualitative research suggesting that top-down (deductive) and bottom-up (inductive) strategies can be integrated during codebook development to balance theoretical grounding with emergent insights [26, 46]. The first two authors developed the codebook, beginning with a top-down strategy guided by established taxonomies of UI design [25] and interaction techniques [27] to define its initial structure and first-level categories. They then applied a bottom-up approach by conducting open coding on a randomly selected subset comprising 50% ($N = 25$) of the papers to refine the first-level categories and inductively develop second-level categories, which served as the final codes for the analysis. This entire process was conducted synchronously, with both authors working together in person, allowing discrepancies to be discussed and a shared understanding of the codebook structure and coding process to be reached.

After finalizing the codes, the first two authors served as coders and independently applied the codebook to the remaining 50% ($N = 25$) of the papers. For each paper, they assigned binary values to denote the presence (1) or absence (0) of each code. Upon completion of the coding process, IRR was calculated using Fleiss' kappa [47] for each individual code. The average kappa score was 0.85, indicating an "almost perfect" level of agreement (0.81-0.99) between the two coders. Two coders also resolved disagreements—primarily concerning whether the use of a slider should be considered a simple parameter adjustment or an interpolative behaviour across multiple variables—through discussion until consensus was reached.

3.5. Design thinking workshop

We conducted a design thinking workshop to propose an interaction model by mapping the findings of our SLR—specifically, the codebooks of interaction techniques and UI design patterns in LLM-based systems that support divergent–convergent thinking—to cognitive tasks observed in CQs elicitation. Additionally, since many CQs elicitation workflows begin with gathering user stories from domain experts, which are later converted into CQs, we therefore also included cognitive tasks present in user story elicitation to inform the model design more comprehensively. This workshop followed the Double Diamond model [14, 42], a widely adopted framework that structures innovation and exploratory analysis into four iterative phases: *Discover*, *Define*, *Develop*, and *Deliver*.

In the *Discover* phase, we reviewed current LLM-based systems for ontology requirements (both user stories and CQs) elicitation—including OntoChat [132, 134], AgOCQs [11], NeOn-GPT [45], RevOnt [35], and RETROFIT-CQs [4]—and gathered experiential insights from researchers involved in ontology requirements engineering to better understand a recurring limitation: fixation [38, 44, 130], where system outputs tend to become overly anchored to users' initial inputs, leading to requirements that fail to adequately reflect the broader ontology scope. Based on this understanding, in the *Define* phase, we identified three critical stages (see Section 4.3) within the divergent–convergent thinking process in requirements elicitation where targeted support is needed to help knowledge engineers mitigate fixation risks. To provide such support, in the *Develop* phase, the first two authors collaboratively mapped the interaction techniques and UI design patterns identified in the codebook to the three critical stages. Through iterative sketching of interaction model prototypes, they co-designed a set of potential interaction models aimed at supporting users' cognitive transitions across these key stages. In the *Deliver* phase, we held structured critique sessions with HCI researchers from the Human-Centred Computing Research Group and OE experts from the Knowledge Graph Research Lab, both at King's College London. Through open feedback discussions and iterative refinement, we reached a consensus on a conceptually grounded and practically feasible interaction model (Section 4.3) that extends the existing LLM-based system OntoChat [132, 134] to support knowledge engineers in requirements elicitation.

4. Findings

After analysing all selected papers, we present the finalized codebooks, which include the codes, their definitions, and referenced evidence for analysing interaction techniques and UI design patterns in LLM-based systems that support divergent–convergent cognitive processes. Detailed explanations of how each code supports these processes are provided in our responses to RQ1 in Section 4.1 and RQ2 in Section 4.2.

4.1. Interaction techniques in LLM-based systems for divergent–convergent thinking

We present the codebook for interaction techniques in Table 2. Our analysis follows Shneiderman's visual information-seeking mantra [99], which emphasizes a sequential process designed to support user cognition—from expansive exploration to focused refinement. We further restructured and elaborated these categories to align with the cognitive process of divergent–convergent thinking.

Categories	Codes	Definition	References	IRR
Spatial navigation	Navigation	Navigation enables users to traverse the design space through directional controls while the LLM dynamically generates region-specific content, helping users build spatial orientation, expand exploratory scope, and construct an initial cognitive map of possibilities.	ABScribe [93], Beyond text-to-image [74], Cells, generators, and lenses [69], Directgpt [79], Genquery [101], IteraTTA [126], Metamorpheus [113], Plantography [60], Promptcharm [118], Promptify [24], Prompting for discovery [8], Snake Story [128], Spellburst [9], Luminate [104], The HaLLMark Effect [59], Wordcraft [131], XCreation [127], CreativeConnect [32], PromptMap [1], Patchview [33], Visar [133], Datatales [105], AutoSpark [31], Lotus [16], Help me write a poem [29], Metaphorian [68], CharacterMeet [91]	0.84
	Zooming	Zooming allows users to adjust the scale of visible information between global and detailed views, with the LLM summarizing high-level clusters and labeling fine-grained elements to support flexible shifting between broad area planning and localized inspection as cognitive demands evolve.	Promptify [24], Prompting for discovery [8], Spellburst [9], Luminate [104], CreativeConnect [32], PromptMap [1], Visar [133], AutoSpark [31]	0.92
Selective attention	Querying	Querying enables users to specify information needs through prompts or keywords, while the LLM suggests semantically relevant items and produces focused summaries, helping users refine attention, sharpen conceptual direction, and progressively narrow the field of exploration.	promptify [24], Prompting for discovery [8], ReelFramer [117], Snake Story [128], Spellburst [9], Luminate [104], Text2Illustration [82], The HaLLMark Effect [59], Where to hide a stolen elephant [100], Wordcraft [131], XCreation [127], Coauthor [72], Story centaur [107], PromptMap [1], Rambler [73], Patchview [33], Sketchar [77], AutoSpark [31], DreamDirector [135], Help me write a poem [29], CharacterMeet [91]	0.76
	Sorting and filtering	Sorting and filtering allow users to reorganize or constrain the visible content based on defined criteria, and the LLM proposes sorting and filtering strategies based on semantic distribution to assist users in prioritizing salient information and reducing cognitive load for structured assessment.	3DALL-E [75], A redhead walks into a bar [49], Beyond text-to-image [74], Breaking barriers to creative expression [108], Cells, generators, and lenses [69], ExpressEdit [111], IteraTTA [126], Plantography [60], PopBlends [115], Promptify [24], Prompting for discovery [8], Luminate [104], The HaLLMark Effect [59], Where to hide a stolen elephant [100], Wordcraft [131], CreativeConnect [32], PromptMap [1], Visar [133], PodReels [116], AutoSpark [31], Metaphorian [68]	0.84
Relational exploration	Linking and brushing	Linking and brushing enable users to select subsets of elements and highlight semantically associated items across views, with the LLM inferring latent connections that help users reason across representations, surface thematic relationships, and build integrated cognitive structures.	3DALL-E [75], A redhead walks into a bar [49], ABScribe [93], Beyond text-to-image [74], Breaking barriers to creative expression [108], Cells, generators, and lenses [69], Directgpt [79], ExpressEdit [111], Plantography [60], PopBlends [115], Promptcharm [118], Promptify [24], Prompting for discovery [8], Snake Story [128], Luminate [104], TaleBrush [34], Text2Illustration [82], The HaLLMark Effect [59], Wordcraft [131], CreativeConnect [32], PromptMap [1], Patchview [33], Visar [133], Datatales [105], AutoSpark [31], Help me write a poem [29], Metaphorian [68]	0.76
	Clustering and combining	Clustering and combining allow users to group or merge related elements into higher-order constructs, and the LLM proposes organizational and consolidation strategies that assist users in externalizing emerging patterns and synthesizing distributed information into coherent conceptual units.	Plantography [60], Promptify [24], Prompting for discovery [8], Luminate [104], CreativeConnect [32], PromptMap [1], AutoSpark [31]	0.88
Details-on-demand	Mouse hover and click-to-expand	Mouse hover and click-to-expand provide lightweight or persistent access to detailed information, with the LLM generating summaries for previews and elaborations for expanded content to help users verify interpretations, critically evaluate alternatives, and integrate insights during resolution.	3DALL-E [75], A redhead walks into a bar [49], ABScribe [93], Beyond text-to-image [74], BunCho [87], CALYPSO [136], Cells, generators, and lenses [69], Co-writing screenplays and theatre scripts with language models [83], Directgpt [79], Genquery [101], IteraTTA [126], LAVE [114], Metamorpheus [113], Promptcharm [118], Promptify [24], Prompting for discovery [8], Snake Story [128], Spellburst [9], Luminate [104], TaleBrush [34], The HaLLMark Effect [59], XCreation [127], Coauthor [72], PromptMap [1], Patchview [33], Visar [133], Datatales [105], AutoSpark [31]	0.72

Table 2
Codebook A: interaction techniques of LLM-based systems for supporting divergent-convergent thinking

Spatial navigation To initiate divergent thinking, users interact with the LLM-based system by first constructing a mental model of the design space to enable broad exploration without prematurely narrowing focus. `Navigation` supports this process by allowing users to traverse the spatial environment using directional controls, such as keyboard arrows, mouse panning, or scroll gestures. As users navigate across multiple regions, the LLM dynamically

generates contextually relevant content based on the area being explored. This interaction enables users to scan the spatial distribution of content, build a sense of orientation, and identify diverse entry points for idea generation. As the visual field becomes increasingly populated with generated content, users experience higher cognitive load, making it difficult to maintain clarity and prioritize attention. To manage this complexity, `Zooming` provides a mechanism for users to dynamically adjust the visible scale of information through scroll wheels or zoom sliders. When users zoom out, the LLM summarizes the content into high-level conceptual overviews, exposing clustering patterns. This helps users plan broader exploration without being overwhelmed by individual details, supporting divergent thinking. When users zoom in, the full text content of the relevant items is displayed, and the LLM automatically generates abstract semantic labels to highlight each item's main thematic focus. These labels assist users in quickly scanning and comparing detailed elements without requiring full reading, thus managing cognitive load during targeted inspection. This seamless transition between abstraction levels, supported by LLM-driven summarization, enables users to fluidly shift between expansive idea exploration and focused idea refinement, mirroring the iterative cycles of divergent–convergent thinking.

Selective attention After building an initial understanding of the design space, users interact with the LLM-based system to progressively focus on elements that align with emerging goals or preferences. `Querying` supports this transition by enabling users to articulate specific intentions through natural language prompts, keywords, or structured parameters. In response, the LLM analyses the existing space to suggest semantically relevant items and generates focused summaries that highlight key points of interest. This interaction helps users rapidly identify meaningful directions and narrow their conceptual focus without losing continuity with the broader space. As users work with increasingly targeted content, `Sorting and filtering` further structure the interaction. The LLM, drawing on the semantic distribution of current items, proposes additional sorting or filtering strategies to optimize content organization. For `Sorting`, users select or specify criteria to reorder visible elements, adjusting the structure based on attributes such as recency, relevance, or importance. For `Filtering`, users apply constraints to selectively display subsets of content, minimizing cognitive load and concentrating attention on information matching defined thresholds. By foregrounding high-priority content while preserving overall spatial context, these interactions enable users to transition from broad ideation toward structured synthesis, supporting the convergent phases of creative analysis.

Relational exploration As users narrow the scope of relevant content, they interact with the LLM-based system to synthesize information and uncover deeper relationships within the design space. `Brushing` supports this process by allowing users to select subsets of elements through hovering, clicking, or dragging within a representation. Based on these selections, the LLM analyses the surrounding space and identifies semantically related items, guiding users toward latent associations that extend beyond immediate visual proximity. `Linking` coordinates the interface to highlight corresponding elements across multiple views according to the semantic relationships inferred by the LLM, facilitating multidimensional reasoning and enabling users to trace structural correspondences that may not be evident from a single perspective. Building on these associations, `Clustering` offers mechanisms for structuring higher-order groupings across the content space. The LLM, informed by the spatial distribution and semantic properties of all elements, generates clustering suggestions that externalize underlying thematic structures. These clusters allow users to reorganize content based on emergent patterns rather than surface attributes, supporting deeper synthesis. In contrast, `Combining` focuses on user-selected subsets. Taking these elements as input, the LLM proposes multiple alternative combination strategies, encouraging users to explore different ways of consolidating related items. This interaction enables comparative evaluation and supports both divergent and convergent cognitive phases: divergence by promoting the exploration of multiple organization pathways, and convergence by guiding the integration of content into coherent conceptual units. Through this layered progression—from selection and association to organization and integration—relational exploration supports users' transition from information gathering toward structured synthesis, supporting iterative cycles of creative analysis.

Details-on-demand In the later stages of interaction, users often require specific information to verify, refine, or synthesize content. At this point, cognitive focus narrows, yet maintaining awareness of the broader context remains essential. `Mouse hover` and `Click-to-expand` support this phase by providing on-demand access to detailed information without disrupting the current view. `Mouse hover` enables transient previews when the

cursor rests over an element. Based on the local content, the LLM generates concise summaries or key attributes that populate these previews, helping users quickly grasp relevant details with minimal cognitive interruption. In contrast, `Click-to-expand` offers persistent access to elaborated content through explicit user input, such as clicking a visual element or icon. Upon expansion, the LLM generates in-depth explanations or contextual annotations derived from the selected item, supporting users in critical evaluation and deeper synthesis. Through this layered access to information, supported by LLM-generated summaries and elaborations, users can selectively deepen their engagement without losing track of the surrounding structure. This interaction flow supports the convergent process of confirming interpretations, refining outputs, and integrating insights within the evolving creative analysis.

4.2. UI designs in LLM-based systems for divergent–convergent thinking

We present the codebook of UI designs in Table 3. Our analysis follows the UI design taxonomy proposed by Braham et al. [25] and further elaborates on its sub-categories. To support this analysis, we include illustrative sketches of different UI designs in Figure 2. Since our SLR focuses on creativity support systems, many UI designs are tailored to specific artistic domains, such as music composition or storytelling. To enhance cross-domain applicability, we abstract these domain-specific design patterns and analyse the transferable patterns that can support divergent–convergent thinking across varied contexts.

Conversation The `chatbot` UI (Figure 2a) is a vertically scrollable conversational interface that enables turn-based interaction through natural language [61, 106, 136]. It structures user engagement into sequential, manageable exchanges, helping users process information incrementally and focus on one idea at a time rather than navigating the entire conceptual space [29, 77, 82, 131]. The LLM assists this process by generating contextually relevant responses and suggesting follow-up options based on user input, which helps maintain conversational flow and guide users toward progressive refinement. Quick-reply buttons present constrained choices, prioritized by the LLM based on recent dialogue context, reducing decision complexity and directing attention to actionable refinements [135]. The persistent scrollable history allows users to revisit previous turns, reinforcing continuity in reasoning and supporting transparent iterative development [59, 106].

Sketching, drawing, and painting The `canvas` UI (Figure 2b) is a free-form graphical interface that supports open-ended visual composition across a continuous two-dimensional workspace [9, 34]. Users create, arrange, and manipulate elements such as lines, shapes, text, and annotations without the constraints of predefined structures, encouraging associative thinking and parallel ideation [104, 127]. To support this exploration, the LLM provides optional suggestions for content fragments or semantic labels based on the evolving spatial distribution, helping users recognize emerging patterns without interrupting creative flow. Toolbars and drawing tools facilitate rapid externalization of mental imagery, while the LLM complements this process by proposing contextually relevant prompts when users appear to stall, thus helping to sustain creative momentum with minimal cognitive load [34].

Collage and collection The `moodboard` UI (Figure 2c) is a collage-based workspace designed for collecting and arranging heterogeneous visual materials, such as images, colours, textures, and text, without fixed structure [28, 32, 68]. It supports fluid exploration by allowing users to juxtapose diverse elements through flexible drag-and-drop manipulation, encouraging associative thinking and the parallel emergence of multiple thematic directions [60, 101, 133]. To assist in organizing the evolving layout, the LLM analyses spatial groupings and can suggest emergent clusters or prompt users with semantic tags that reflect latent connections between collected items. By visually clustering related content, users externalize intuitive patterns and branches of ideas, while organizational tools like alignment guides and asset panels reduce layout complexity and help maintain creative momentum [32, 33].

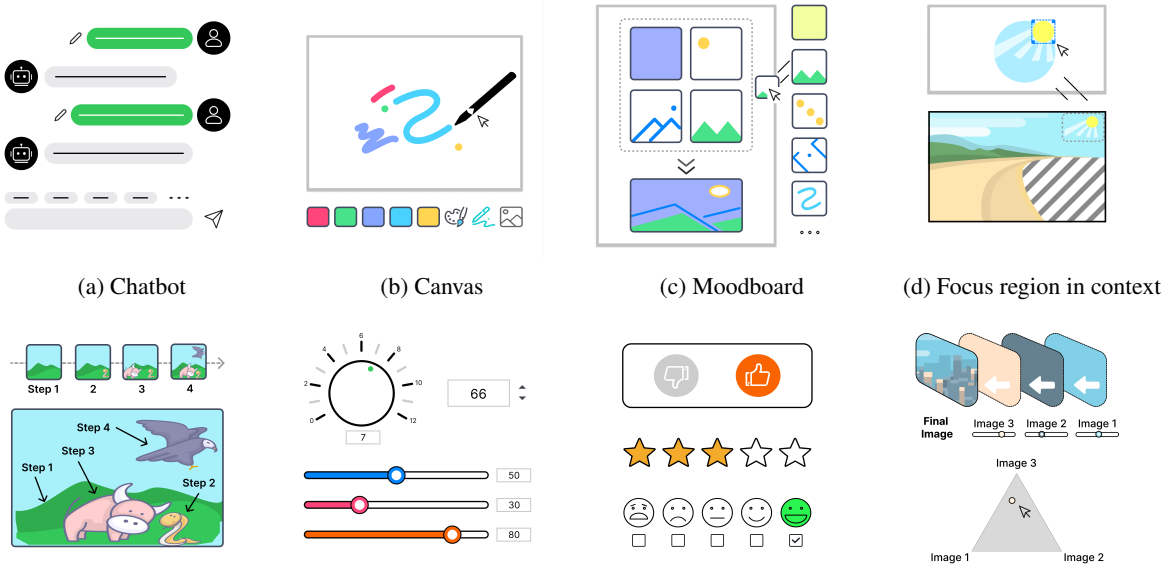
Divide-and-Conquer and Content Infilling The `focus region in context` UI (Figure 2d) and the `spatially structured or temporally guided editor` UI (Figure 2e) are designed to support focused refinement without losing awareness of broader structures [79, 113]. The `focus region in context` UI provides a composite view linking a zoomed-in editable area with an overview of the full workspace, facilitating divide-and-conquer strategies by enabling localized editing while preserving global coherence [59, 60]. As users

Categories	Codes	Definition	References	IRR
Conversation	Chatbot	Chatbot structures interaction into manageable conversational turns where the LLM generates contextually relevant replies and suggests next prompts, supporting users' cognitive flow by segmenting complex thinking into incremental steps and maintaining attention on immediate reasoning goals.	CALYPSO [136], GenAssist [61], I spy a metaphor [30], Language as reality [106], LAVE [114], Snake Story [128], Luminate [104], Text2Illustration [82], The HaLLMark Effect [59], Wordcraft [131], Sketchar [77], DreamDirector [135], Help me write a poem [29], CharacterMeet [91]	0.92
Sketching, drawing, and painting	Canvas	Canvas provides a flexible drawing space where the LLM proposes spatial annotations or refinements based on content evolution, assisting users in externalizing fragmented ideas while promoting the progressive stabilization of emerging mental representations.	Spellburst [9], Luminate [104], TaleBrush [34], XCreation [127]	0.84
Collage and collection	Moodboard	Moodboard enables open arrangement of heterogeneous materials, while the LLM identifies implicit clustering patterns and suggests semantic labels, helping users organize fragmented inputs into coherent visual groupings that support associative reasoning and thematic abstraction.	DesignAID [28], Directgpt [79], Genquery [101], Plantography [60], Promptcharm [118], Promptify [24], CreativeConnect [32], PromptMap [1], Patchview [33], Visar [133], AutoSpark [31], Lotus [16], Metaphorian [68]	0.88
Divide-and-conquer and content infilling	Focus region in context	Focus region in context links local detail editing to global overviews, with the LLM suggesting context-sensitive refinements, helping users maintain cognitive coherence across hierarchical levels while shifting focus between local operations and global understanding.	Directgpt [79], ExpressEdit [111], Plantography [60], Promptcharm [118], Luminate [104], The HaLLMark Effect [59], XCreation [127]	0.76
	Spatially structured and sequentially guiding editor	Spatially structured or temporally guided editors externalize task organization through grids or timelines, where the LLM proposes context-aligned completions, reducing planning overhead and enabling incremental construction of orderly, temporally coherent representations.	3DALL-E [75], Beyond text-to-image [74], Breaking barriers to creative expression [108], Cells, generators, and lenses [69], Co-writing screenplays and theatre scripts with language models [83], ExpressEdit [111], Metamorpheus [113], Luminate [104], TaleBrush [34], The HaLLMark Effect [59], Where to hide a stolen elephant [100], Coauthor [72], Visar [133], PodReels [116], Sketchar [77], DreamDirector [135], Metaphorian [68]	0.84
Tweaking	Slider and knob and numeric textfield	Sliders, knobs, and numeric textfields allow users to vary precision in parameter control, while the LLM generates real-time semantic previews, supporting users in forming accurate mental models of how subtle adjustments affect broader system behavior.	BunCho [87], Cells, generators, and lenses [69], Co-writing screenplays and theatre scripts with language models [83], ExpressEdit [111], PopBlends [115], Promptcharm [118], Promptify [24], Spellburst [9], Wordcraft [131], Story Centaur [107], Rambler [73], Lotus [16]	0.96
Rating	Binary and Likert-scale rating	Binary and Likert-scale rating compress subjective judgments into fixed scales, and the LLM clarifies ambiguous choices through micro-explanations, reducing cognitive ambiguity and helping users quickly consolidate evolving impressions into explicit evaluative decisions.	Help me write a poem [29]	0.92
Interpolation	Interpolating slider, region, and graph	Interpolating controls let users blend properties along continuous dimensions, with the LLM generating interpolated semantic snapshots, aiding users in mentally traversing variant spaces and reinforcing comparative reasoning during gradual preference formation.	Cells, generators, and lenses [69], Plantography [60], Patchview [33]	0.76
Selection and exploration	List and grid	List and grid layouts structure item presentation systematically, while the LLM highlights contextually pertinent candidates, facilitating user scanning processes by focusing selective attention and reducing information overload.	3DALL-E [75], Beyond text-to-image [74], Breaking barriers to creative expression [108], BunCho [87], Cells, generators, and lenses [69], DesignAID [28], IteraTTA [126], LAVE [114], Promptify [24], Prompting for discovery [8], Luminate [104], Text2Illustration [82], CreativeConnect [32], PromptMap [1], Datatales [105], PodReels [116], AutoSpark [31], DreamDirector [135], Lotus [16], Help me write a poem [29]	0.84
	In-flow options	In-flow options introduce context-sensitive suggestions during ongoing interaction, where the LLM predicts likely branching opportunities, supporting users' real-time associative thinking and minimizing interruption in ideation flow.	A redhead walks into a bar [49], ABScribe [93], CALYPSO [136], Cells, generators, and lenses [69], Promptcharm [118], Promptify [24], Prompting for discovery [8], Spellburst [9], Luminate [104], The HaLLMark Effect [59], Wordcraft [131], XCreation [127], Patchview [33], Visar [133], Sketchar [77], DreamDirector [135], Help me write a poem [29], Metaphorian [68], CharacterMeet [91]	0.84
	Node-based branches	Node-based branch UIs externalize conceptual networks spatially, with the LLM suggesting semantic expansions or optimizations, helping users externalize non-linear thought paths and systematically support evolving idea hierarchies.	Cells, generators, and lenses [69], Plantography [60], Spellburst [9], Luminate [104], PromptMap [1], Patchview [33], Visar [133], AutoSpark [31]	0.92
	Map and earth	Map and earth UIs enable spatial exploration at multiple scales, where the LLM highlights semantically promising distant regions, assisting users in reorienting attention to overlooked conceptual territories without losing broader spatial organization.	A redhead walks into a bar [49], Promptify [24]	0.96
Sequence editing	Timeline, storyline, and step sequence	Timeline, storyline, and step sequence UIs present ordered content structures, where the LLM suggests chronological improvements or missing transitions, supporting users' incremental refinement of narrative logic and temporal cohesion.	ExpressEdit [111], IteraTTA [126], LAVE [114], Metamorpheus [113], TaleBrush [34], The HaLLMark Effect [59], Visar [133], DreamDirector [135], Lotus [16], CharacterMeet [91]	0.84
	Lanes and tracks	Lanes or tracks layer multiple streams across shared time axes, where the LLM proposes cross-stream alignments, helping users coordinate concurrent developments and maintain cognitive coherence across interdependent sequence layers.	LAVE [114], PodReels [116], Lotus [16]	0.88

Table 3

Codebook B: UI design of LLM-based systems for supporting divergent-convergent thinking

zoom into specific regions, the LLM supports refinement by suggesting localized adjustments or semantic annotations based on the current content and its surrounding context. Overview thumbnails, bounding boxes, and synchronized navigation controls further assist users in managing transitions between detailed editing and high-level orientation [79, 104, 127]. In parallel, the `spatially structured and sequentially guiding editor` provides predefined spatial or temporal layouts, such as grids, timelines, or segmented tracks [77, 126, 133], which



(a) Chatbot (b) Canvas (c) Moodboard (d) Focus region in context

(e) Spatially structured and sequentially guiding editor (f) Slider and knob and numeric textfield (g) Binary and Likert-scale rating (h) Interpolating slider and region and graph

(i) List and grid (j) In-flow options (k) Node-based branches (l) Map and earth

(m) Timeline and storyline and step Sequencer (n) Lanes and tracks

Fig. 2. Illustrative UI sketches for an LLM-based system supporting divergent–convergent thinking

externalize task-relevant constraints to guide incremental composition or revision. Here, the LLM enhances user workflows by proposing context-aware completions or temporal refinements aligned with the structure, supporting users in maintaining logical flow and minimizing cognitive effort during detailed, sequential editing [72, 113].

Tweaking The `slider, knob, and numeric textfield` UI designs (Figure 2f) provide mechanisms for adjusting parameters with varying levels of precision, supporting users in progressively refining their outputs [69, 83, 107]. The `slider` offers a continuous scale for coarse adjustments by dragging a handle, helping users

rapidly explore broader value ranges without focusing on exact numbers [24]. The `knob` enables finer-grained control over discrete values, allowing users to incrementally tune parameters while maintaining a sense of relative positioning [69]. The `numeric textfield` supports exact input of arbitrary values, including decimals, for situations requiring precise specification [87, 111]. During adjustment, the LLM dynamically generates real-time previews or contextual explanations based on current parameter states, helping users understand the semantic impact of changes and focus attention on meaningful refinements. This combination of interaction techniques supports an initial phase of broad adjustment followed by progressive narrowing, enabling users to balance exploration and precision as cognitive demands shift toward outcome optimization.

Rating The `binary and Likert-scale rating` UI designs (Figure 2g) provide predefined response scales to efficiently capture user preferences [29]. A binary rating presents two exclusive options, such as yes/no or like/dislike, while a Likert-scale rating offers an ordinal range, typically from 1 to 5, through easily selectable interfaces like radio buttons or stars. In this setting, the LLM can pre-annotate options with brief clarifications or examples derived from the current context, helping users disambiguate meanings and make faster, more confident choices. By compressing subjective evaluation into constrained selections, these rating interfaces reduce decision complexity, enable rapid comparison across alternatives, and help users crystallize evolving judgments into stable preferences as cognitive focus converges toward decision closure.

Interpolation The `interpolating slider, region, and graph` UI designs (Figure 2h) provide continuous control mechanisms for exploring variations across multiple options [60]. The `interpolating slider` enables movement along a one-dimensional axis to transition smoothly between representative points, while the `region`-based interface generalizes interpolation to two dimensions, blending properties from multiple anchors based on dragging positions [69]. The `graph`-based interface presents variables as curves or nodes, allowing users to fine-tune weighted contributions through direct manipulation [33]. To support this exploration, the LLM can generate semantic previews or interpolated examples at different points along the control surface, helping users better understand emerging alternatives without needing to test every option manually. These components encourage continuous experimentation and promote flexible mental navigation across possibilities, supporting users in discovering viable directions while maintaining creative fluidity.

Selection and Exploration The `list and grid` UI designs (Figure 2i) organize options into structured layouts, facilitating broad scanning and comparison [1, 82]. Lists present items sequentially, while grids arrange them spatially for faster visual contrast [126]. The LLM can enhance these views by dynamically highlighting contextually relevant candidates based on user interaction history, making it easier for users to focus exploration without feeling overwhelmed. The `in-flow options` UI (Figure 2j) introduces lightweight suggestion elements at decision points, where the LLM generates contextual prompts or micro-choices derived from the current creative trajectory [9, 77]. This lowers interruption cost and helps users branch into new ideas seamlessly. The `node-based branches` UI (Figure 2k) uses editable graphs to represent content or logic, allowing users to arrange nodes spatially and expand conceptual paths [60]. Here, the LLM can propose possible node expansions or missing connections, encouraging users to pursue alternative developments that might otherwise remain unexplored. Lastly, the `map and earth` UI (Figure 2l) provides spatial overviews where the LLM can annotate or highlight promising areas based on semantic clustering of prior content, helping users identify distant but potentially fruitful regions for further discovery [24]. Together, these selection and exploration interfaces support wide-ranging ideation by organizing choices clearly, introducing branching opportunities dynamically, and reinforcing users' cognitive flexibility during early-stage exploration.

Sequence Editing The `timeline, storyline, and step sequence` UI designs (Figure 2m) are sequence-based interfaces that arrange events, content, or actions along a linear progression [111, 133]. Timelines and storylines display elements ordered across continuous or discrete time, while step sequences divide timelines into uniform intervals where users can assign specific content or triggers [59, 111]. The LLM can assist during sequence construction by suggesting temporal insertions, anticipating missing steps, or recommending sequence refinements based on the evolving structure, helping users maintain coherence without manual micromanagement. These interfaces support convergent cognitive processes by guiding users to incrementally adjust temporal order, resolve inconsistencies, and refine detail placement within an evolving linear framework. The `lanes or tracks`

UI (Figure 2n) extends this sequencing principle by layering multiple parallel streams—such as voice, animation, or background sound—along a shared time axis [16, 114]. Each lane represents an independent dimension, allowing users to edit horizontally across time and vertically across layers while maintaining logical separation. The LLM can enhance this process by proposing cross-track alignments, highlighting synchronization mismatches, or recommending compositional improvements across layers based on inferred narrative or structural goals. These supports externalize complex dependencies and reduce cognitive overload during detailed adjustment, enabling users to progressively converge on synchronized, coherent outputs across multiple interrelated sequences.

4.3. Supporting knowledge engineers in ontology requirements elicitation through interaction design with LLMs

To support knowledge engineers in the iterative process of CQs elicitation—aimed at progressively auditing ontology boundaries—we map interaction techniques and UI designs identified from LLM-based creative systems to cognitive tasks aligned with divergent–convergent thinking patterns in ontology requirements (both user stories and CQs) elicitation, as detailed through a design thinking workshop in Section 3.5. Based on this mapping, we propose an interaction model that extends the existing LLM-based system OntoChat [132, 134] to support knowledge engineers in CQs elicitation. In this section and Section 5, we use **`UI`** to refer to user interface designs and **`IT`** to refer to interaction techniques when specifying which interaction design elements identified in the previous two sections are applied in the proposed interaction model.

Before introducing our proposed interaction model, we first outline the interaction limitations of the existing OntoChat system [132, 134]. In its current form, OntoChat is built around a `chatbot UI`, where an LLM emulates the role of a knowledge elicitor by guiding domain experts through prompt-driven conversations to elicit and refine ontology user stories, which are then automatically converted into CQs. While the system is intended to foster divergent thinking—by offering follow-up questions and varied suggestions—it frequently leads users prematurely into convergent thinking. This issue arises from the chatbot's inherently turn-based and linear interaction structure, which encourages users to respond directly to each prompt in sequence, prompting progressive narrowing of ideas and making it difficult for users to step back with an all-encompassing view to consider alternative directions. As a result, the `chatbot UI` causes fixation [38, 44, 63, 104, 130], where users become anchored to their initial inputs and fail to generate a sufficiently diverse and representative set of user stories, which in turn leads to extracted CQs that inadequately reflect the scope of the intended ontology. To help users avoid fixation, we propose a new interaction model for OntoChat to support requirements (CQs) elicitation. We choose CQs as the form of requirements because they are one of the most commonly used techniques in ontology requirements engineering, and their direct elicitation helps avoid the uncertainty—such as misinterpretation of intent—introduced by LLM-based pipelines when converting user stories into CQs. The proposed interaction model is detailed in the following sections.

Initialization of the space of CQs Referring to Figure 3, the elicitation process begins when a knowledge engineer submits a prompt through a `chatbot UI` to specify the ontology's target application domain. In response, the system uses an LLM to generate an initial set of candidate CQs, each paired with an answer (a single terminology it suggested). For each pair, the LLM also assigns a *domain coverage* label that indicates the subdomain to which the CQ's answer belongs (e.g., body systems, symptoms, diagnostic tools). These CQs are displayed on a `canvas UI` structured by two key dimensions: *domain coverage* and *terminology granularity*. The latter reflects the specificity of the answer, which is computed by the system based on the path length from the answer (a single terminology) to the root node *entity* in WordNet [81]. Each CQ is positioned on the canvas according to its values in these two dimensions, forming a spatial overview of how the elicited CQs are distributed. Knowledge engineers can explore this layout using `navigation IT` controls such as panning or scrolling and manage cognitive complexity with `zooming IT` to shift between high-level overviews and detailed views. This spatial structure, combined with these interaction techniques, supports iterative exploration and refinement of the ontology scope.

CQs elicitation through divergent–convergent thinking After the initial canvas has been populated with candidate CQs, the system supports a structured process of iterative CQs elicitation based on divergent–convergent thinking. This process allows knowledge engineers to explore the conceptual scope broadly and then gradually refine it to achieve more precise coverage of the target ontology. In the divergent phase, the system supports dimension-guided

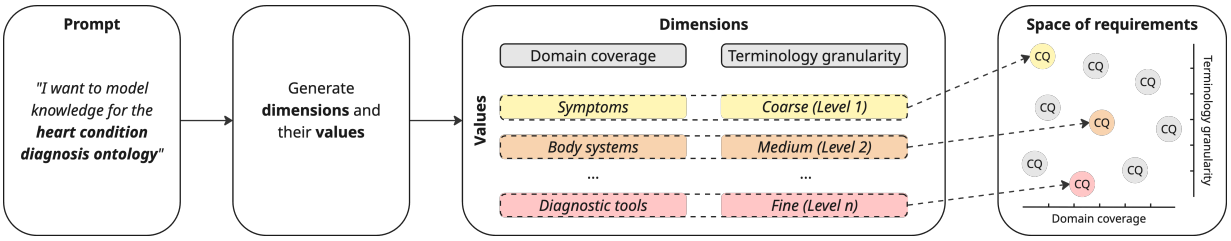


Fig. 3. Initialization of the space of CQs

CQ augmentation, allowing knowledge engineers to click a button to expand the set of CQs within a specific region of the `canvas UI` (Figure 4a). This triggers the LLM to generate new CQs that share the same dimension values—namely, domain coverage and terminology granularity—as the selected region. This mechanism enables localized expansion of the CQ space without disturbing its overall structure, supporting a focused yet systematic elaboration of underdeveloped areas. To further enhance divergence, the system also supports dimension values augmentation (Figure 4b). Knowledge engineers can define new values for the organizing dimensions—either domain coverage or terminology granularity—or request the LLM to suggest them. The `in-flow options UI` supports this process by offering dynamic suggestions for new dimension values for knowledge engineers to select. These options are presented directly within the interaction flow, minimizing disruption to the knowledge engineer's creative thought process. When a new value is selected, the `canvas UI` layout is updated accordingly to incorporate the newly underexplored regions. Knowledge engineers may then click these regions to instruct the LLM to generate new CQs that correspond to the combination of the selected domain and granularity levels. This guided divergence leads to a more evenly distributed set of CQs that reflect the conceptual diversity intended for the target domain.

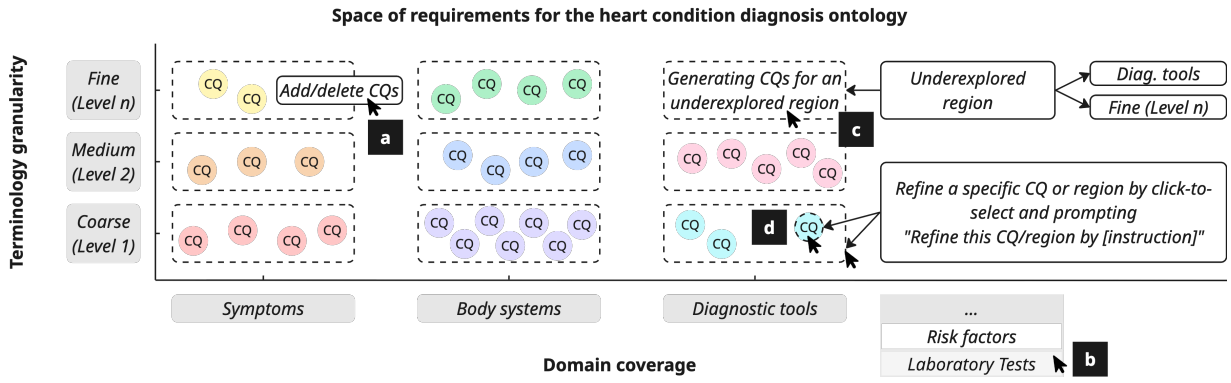


Fig. 4. CQs elicitation through divergent–convergent thinking

As the number of generated CQs increases, visual overload can occur, making navigation more challenging. To address this, the system supports `focus region in context UI` through `zooming IT` (Figure 5), enabling knowledge engineers to switch between levels of detail based on their current focus. This is achieved through scrolling on a `slider UI` or by clicking, giving knowledge engineers fine-grained control over how they allocate their attention across the `canvas UI`. At the 1× zoom level, CQs are represented as minimal visual markers to indicate distribution without revealing text, similar to how `node-based branches UI` use nodes to represent simplified concepts. Knowledge engineers can also view the full text of a specific CQ and its answer by using `click-to-expand IT`, where clicking on a marker temporarily reveals detailed information without disrupting the overall layout. This broad overview, combined with lightweight access to details, supports divergent thinking by allowing knowledge engineers to explore the overall space of the ontology without text overload while encouraging them to make informed decisions about where to converge for further exploration. After the knowledge engineer decides where to converge, they can zoom in to the 2× level, where the terminology (i.e., the answer to each CQ)

for the narrowing regions is displayed. Knowledge engineers can then select the region they are most interested in and zoom in further to the 3× level, where they can view the LLM-generated semantic summary of CQs in that region. Knowledge engineers can use this summary to initially assess whether CQ augmentation or refinement is needed in the selected region. If so, they can zoom in to the 4× level, where the full text of each CQ–answer pair is displayed, allowing knowledge engineers to audit and perform further CQ refinement in that region, if needed. Overall, this zooming in process supports convergent thinking by enabling knowledge engineers to focus on specific aspects of the CQ space and continuously narrow their scope toward targeted refinement. At the same time, knowledge engineers can shift between zooming in and out, facilitating a fluid transition between expansive exploration and focused review. This flexibility supports both divergent thinking and convergent thinking processes, allowing knowledge engineers to adapt their approach as needed.

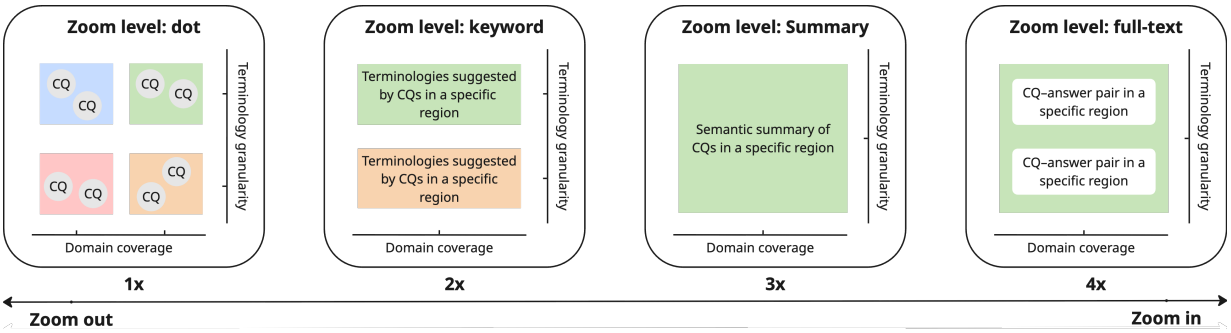


Fig. 5. Zooming

Except for auditing specific CQs or regions each time, this system also supports refining a group of CQs or regions to clarify the boundaries of the target domain more efficiently. Knowledge engineers can utilize the system's `querying IT` and `filtering IT` functions by entering keywords to highlight a `focus region in context UI`, where the queried or filtered results are visually emphasized on the `canvas UI` (Figure 6). In addition to direct matches, the LLM analyses the query to identify semantically aligned CQs, highlighting them to support broader conceptual retrieval. Unselected content is not removed but instead has its visual salience adjusted, allowing knowledge engineers to focus their attention and reduce visual complexity while maintaining awareness of the overall structure. The queried and filtered results are also presented using a `list and grid UI`, enabling knowledge engineers to select a subset of CQs they are most interested in for further analysis. To support cross-referencing, the system enables `linking and brushing IT`, where selecting one or multiple results in the list highlights only the corresponding content on the `canvas UI`. This interaction makes it easier for knowledge engineers to compare the most relevant results and organize their refinement efforts more efficiently. Once a desired group of CQs or regions is selected, the system enables context-aware CQ refinement (Figure 3d). Knowledge engineers can issue refinement instructions through the `chatbot UI` interface. The system then constructs an input to the LLM by combining the user's prompt with the current state of the selected CQs and their spatial context on the canvas. Based on this context, the LLM generates CQs remain consistent with the ontology's evolving scope. This process continues iteratively until the knowledge engineer is satisfied with the defined scope.

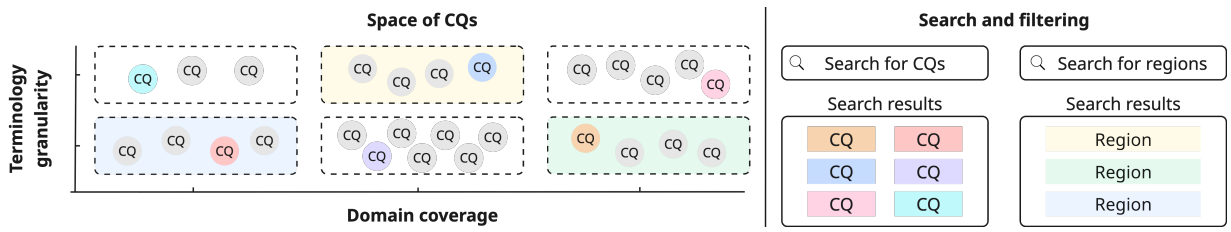


Fig. 6. Search and filtering

In summary, the new interaction model for OntoChat supports knowledge engineers throughout the CQs elicitation process by aligning interaction design with divergent–convergent thinking. The initial canvas, structured by domain coverage and terminology granularity, provides an organized view of the space of CQs, reflecting the ontology's evolving scope. This new interaction model empowers knowledge engineers to iteratively explore and refine the scope in a structured, traceable manner, supporting the progressive auditing of ontology boundaries and ultimately leading to more effective coverage of the target domain.

5. Discussion

5.1. Transferability of interaction design from the arts and creativity domain to the OE domain

While artistic creation and ontology requirements elicitation pursue different end goals—creative expression versus structured knowledge representation—they share a common cognitive foundation rooted in divergent–convergent thinking [41, 104]. This shared cognitive process provides the basis for examining how interaction design originally developed for creativity support systems can be meaningfully adapted to ontology requirements elicitation systems.

In particular, the transferability of interaction design can be understood through their support for divergent and convergent thinking processes. To support divergent thinking, interaction techniques in the creativity domain such as `navigation IT` and `zooming IT`, together with corresponding UI designs such as `canvas UI`, `slider UI`, `in-flow options UI`, and `node-based branches UI`, enable designers to move freely across large visual spaces, explore alternative creative directions across dimensions such as theme, composition, and style, and uncover new stylistic possibilities without premature commitment. Similarly, in ontology requirements elicitation, these interaction techniques and UI designs allow knowledge engineers to broadly explore the space of candidate CQs organized along *domain coverage* and *terminology granularity* dimensions. This spatial exploration supports dimensional reasoning [5, 17], supporting knowledge engineers in identifying underexplored regions, comparing clusters of CQs across dimensions, and progressively developing an informed understanding of the ontology's evolving scope without prematurely constraining its boundaries.

To support convergent thinking, interaction techniques in the creativity domain such as `querying IT`, `filtering IT`, `click-to-expand IT`, and `linking and brushing IT`, together with corresponding UI designs such as `chatbot UI`, `focus region in context UI`, `slider UI`, and `list and grid UI`, enable designers to retrieve specific assets, compare alternatives, and selectively refine elements to converge on a coherent creative outcome while maintaining a view of the broader creative landscape. In ontology requirements elicitation, the same interaction design allow knowledge engineers to retrieve specific CQs, compare the concepts suggested by retrieved CQs, and selectively refine CQs to improve boundary precision while retaining an awareness of the evolving ontology scope. This balance between local focus and global context ensures that detailed refinements remain aligned with the ontology's conceptual boundaries, supporting coherent and traceable scope development.

However, it is important to recognize that not all interaction design from the arts and creativity domain are appropriate for adaptation to ontology requirements elicitation. For example, `lanes and tracks UI` are designed for managing parallel streams of time-based media, where each lane typically represents an independent audio track or a sequence of video frames. This structure is not directly applicable to ontology requirements elicitation, as the space of requirements does not naturally involve multiple parallel timelines. Nevertheless, the underlying principle of maintaining versioned development over time remains valuable. In ontology requirements elicitation, a similar UI design could be adapted to track successive versions of the evolving space of requirements within a single timeline, enabling knowledge engineers to revisit earlier modelling decisions, compare alternative development paths, and make informed choices about where to diverge or converge in order to progressively auditing the boundaries of the ontology.

5.2. Towards implementation and evaluation

The proposed interaction model synthesizes a set of interaction techniques and UI designs that align with divergent–convergent thinking during CQs elicitation. To move beyond conceptual specification, the model must be instantiated through a system prototype and empirically validated in domain-relevant settings. A practical implementation path involves extending OntoChat [132] with key components from the model: a spatial canvas for organizing candidate CQs by domain coverage and terminology granularity, interactive zooming for detail management, and refinement workflows guided by querying and brushing. These components can be embedded within a cohesive interface to support controlled user testing. We propose a multi-method evaluation strategy. First, an A/B study comparing the enhanced prototype with the baseline OntoChat can reveal the model's effects on fixation reduction, measured through semantic diversity and domain breadth of elicited CQs. Second, think-aloud sessions with knowledge engineers can provide insight into how the spatial layout and interaction mechanisms influence cognitive navigation between idea generation and refinement. Third, task-based evaluations can incorporate cognitive load instruments, such as NASA-TLX [56], to assess whether the model mitigates overload during iterative elicitation. Fourth, qualitative interviews can examine how users perceive the model's support for revisiting and restructuring elicited content. Lastly, an expert-led analysis of CQs coverage can assess whether the elicited CQs better reflect the intended scope of the target ontology. Together, these evaluations will determine the model's capacity to support systematic exploration and synthesis during knowledge modelling tasks.

5.3. Generalizability of the proposed interaction model

Although our proposed interaction model was developed for CQs elicitation in OE, its core design principles—organizing ideas along meaningful dimensions, supporting visual navigation, and enabling iterative expansion and refinement—can be applied to many other knowledge-intensive and creative domains. In writing, for example, authors often work with a large number of narrative elements. These can be organized along dimensions such as character development (from background roles to protagonists) and story arc (from exposition to climax), allowing writers to explore alternative plot structures before refining them into a final draft. In game design, developers balance dimensions like mechanics complexity and user engagement, placing game elements in a design space to prototype various player experiences. In speculative design, designers might explore a range of social or technological futures by combining dimensions such as societal impact and technological feasibility, enabling divergent exploration of provocative concepts followed by convergence on plausible scenarios. Similarly, VR scene creation often involves positioning elements in a 3D space while considering narrative function and spatial interaction zones, where designers need to continuously switch between high-level layout and fine-grained object placement. In all of these cases, the ability to visually organize, compare, and refine ideas—supported by interaction techniques like zooming, filtering, and semantic grouping—mirrors the cognitive workflow supported by our model.

6. Limitations

This study has several limitations. First, while the proposed interaction model supports knowledge engineers in auditing the space of CQs, the quality of the generated CQs is evaluated only in terms of boundary completeness. Other important aspects of CQs quality—such as feasibility, clarity, or logical consistency—are not addressed within the current system. Integrating mechanisms for multi-dimensional quality assessment remains an important direction for future work. Second, considering the rapid expansion of LLM-related publications, our SLR dataset is constrained by time and scope, so the review may not capture the full breadth of relevant systems. Moreover, our synthesis focused specifically on interaction patterns that explicitly support divergent–convergent thinking in art and creativity. This choice may have omitted techniques embedded in general-purpose LLM systems that were not framed under this cognitive model. As a result, while the design space we present is representative of current creative-system interactions, it may not be exhaustive.

7. Conclusions

In this paper, we examined how interaction design can enhance CQs elicitation in OE by leveraging the cognitive framework of divergent–convergent thinking. Through a SLR of 50 LLM-based systems in creative domains, we developed a taxonomy of interaction techniques and UI patterns that support iterative exploration and refinement. These findings informed a novel interaction model that extends the existing OntoChat system, enabling knowledge engineers to spatially navigate candidate CQs, reason about their distribution along dimensions such as domain coverage and terminology granularity, and iteratively expand, refine, and audit the scope of the target ontology. The proposed model offers a reusable design scaffold for future human–LLM systems, particularly in tasks requiring human judgment to surface, evaluate, and reorganize latent conceptual structures. While developed for CQs elicitation, the model has broader relevance to other OE and creative tasks, such as ontology alignment, hierarchical refinement, or scenario planning. Future work will focus on implementing and evaluating the interaction model's usability, cognitive load, and contribution to CQs quality in OntoChat.

References

[1] K. Adamkiewicz, P.W. Woźniak, J. Dominiak, A. Romanowski, J. Karolus and S. Frolov, PromptMap: An Alternative Interaction Style for AI-Based Image Generation, in: *Proceedings of the 30th International Conference on Intelligent User Interfaces*, 2025, pp. 1162–1176.

[2] C. Ahlberg and B. Shneiderman, The alphaslider: a compact and rapid selector, in: *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1994, pp. 365–371.

[3] C. Ahlberg, C. Williamson and B. Shneiderman, Dynamic queries for information exploration: An implementation and evaluation, in: *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1992, pp. 619–626.

[4] R. Alharbi, V. Tamma, F. Grasso and T. Payne, An experiment in retrofitting competency questions for existing ontologies, in: *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*, ACM, 2024, pp. 1650–1658.

[5] R. Alharbi, V. Tamma, F. Grasso and T.R. Payne, Investigating Open Source LLMs to Retrofit Competency Questions in Ontology Engineering, in: *Proceedings of the AAAI Symposium Series*, Vol. 4, 2024, pp. 188–198.

[6] R. Alharbi, V. Tamma, F. Grasso and T.R. Payne, A review and comparison of competency question engineering approaches, in: *International Conference on Knowledge Engineering and Knowledge Management*, Springer, 2024, pp. 271–290.

[7] R. Alharbi, V. Tamma, F. Grasso and T.R. Payne, The role of Generative AI in competency question retrofitting, in: *European Semantic Web Conference*, Springer, 2024, pp. 3–13.

[8] S.G. Almeda, J. Zamfirescu-Pereira, K.W. Kim, P. Mani Rathnam and B. Hartmann, Prompting for discovery: Flexible sense-making for ai art-making with dreamsheets, in: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–17.

[9] T. Angert, M. Suzara, J. Han, C. Pondoc and H. Subramonyam, Spellburst: A node-based interface for exploratory creative coding with natural language prompts, in: *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023, pp. 1–22.

[10] M.-J. Antia and C.M. Keet, Assessing and enhancing bottom-up CNL design for competency questions for ontologies, in: *Proceedings of the Seventh International Workshop on Controlled Natural Language (CNL 2020/21)*, 2021.

[11] M.-J. Antia and C.M. Keet, Automating the Generation of Competency Questions for Ontologies with AgOCQs, in: *Iberoamerican Knowledge Graphs and Semantic Web Conference*, Springer, 2023, pp. 213–227.

[12] F. Baader, *The description logic handbook: Theory, implementation and applications*, Cambridge university press, 2003.

[13] B. Bach, Z. Wang, M. Farinella, D. Murray-Rust and N. Henry Riche, Design patterns for data comics, in: *Proceedings of the 2018 chi conference on human factors in computing systems*, 2018, pp. 1–12.

[14] B.H. Banathy, *Designing social systems in a changing world*, Springer Science & Business Media, 2013.

[15] L. Bartram, A. Ho, J. Dill and F. Henigman, The continuous zoom: A constrained fisheye technique for viewing and navigating large information spaces, in: *Proceedings of the 8th annual ACM symposium on User interface and software technology*, 1995, pp. 207–215.

[16] A. Barua, K. Benharrak, M. Chen, M. Huh and A. Pavel, Lotus: Creating Short Videos From Long Videos With Abstractive and Extractive Summarization, in: *Proceedings of the 30th International Conference on Intelligent User Interfaces*, 2025, pp. 967–981.

[17] M. Beaudouin-Lafon and W.E. Mackay, Prototyping tools and techniques, in: *The human-computer interaction handbook*, CRC Press, 2007, pp. 1043–1066.

[18] B.B. Bederson and J.D. Hollan, Pad++ a zooming graphical interface for exploring alternate interface physics, in: *Proceedings of the 7th annual ACM symposium on User interface software and technology*, 1994, pp. 17–26.

[19] A. Bezerianos, F. Chevalier, P. Dragicevic, N. Elmqvist and J.-D. Fekete, Graphdice: A system for exploring multivariate social networks, in: *Computer graphics forum*, Vol. 29, Wiley Online Library, 2010, pp. 863–872.

[20] C. Bezerra, F. Santana and F.L.G. de Freitas, CQChecker: A Tool to Check Ontologies in OWL-DL using Competency Questions written in Controlled Natural Language, *Learning and Nonlinear Models* **12** (2014), 115–129. https://api.semanticscholar.org/CorpusID:55282107.

[21] M.M. Biskjaer, P. Dalsgaard and K. Halskov, A constraint-based understanding of design spaces, in: *Proceedings of the 2014 conference on Designing interactive systems*, 2014, pp. 453–462.

[22] J.O. Borchers, A pattern approach to interaction design, in: *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques*, 2000, pp. 369–378.
[23] N. Boukhelifa, A. Bezerianos, I.C. Trelea, N.M. Perrot and E. Lutton, An exploratory study on visual exploration of model simulations by multiple types of experts, in: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–14.
[24] S. Brade, B. Wang, M. Sousa, S. Oore and T. Grossman, Promptify: Text-to-image generation through interactive prompt exploration with large language models, in: *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023, pp. 1–14.
[25] A. Braham, F. Buendía, M. Khemaja and F. Gargouri, User interface design patterns and ontology models for adaptive mobile applications, *Personal and Ubiquitous Computing* **26**(6) (2022), 1395–1411. doi:10.1007/s00779-020-01481-5.
[26] V. Braun and V. Clarke, Using thematic analysis in psychology, *Qualitative research in psychology* **3**(2) (2006), 77–101.
[27] F. Brudy, C. Holz, R. Rädle, C.-J. Wu, S. Houben, C.N. Klokmose and N. Marquardt, Cross-device taxonomy: Survey, opportunities and challenges of interactions spanning across multiple devices, in: *Proceedings of the 2019 chi conference on human factors in computing systems*, 2019, pp. 1–28.
[28] A. Cai, S.R. Rick, J.L. Heyman, Y. Zhang, A. Filipowicz, M. Hong, M. Klenk and T. Malone, DesignAID: Using generative AI and semantic diversity for design inspiration, in: *Proceedings of The ACM Collective Intelligence Conference*, 2023, pp. 1–11.
[29] T. Chakrabarty, V. Padmakumar and H. He, Help me write a poem: Instruction tuning as a vehicle for collaborative poetry writing, *arXiv preprint arXiv:2210.13669* (2022).
[30] T. Chakrabarty, A. Saakyan, O. Winn, A. Panagopoulou, Y. Yang, M. Apidianaki and S. Muresan, I spy a metaphor: Large language models and diffusion models co-create visual metaphors, *arXiv preprint arXiv:2305.14724* (2023).
[31] L. Chen, Q. Jing, Y. Tsang, Q. Wang, R. Liu, D. Xia, Y. Zhou and L. Sun, AutoSpark: Supporting Automobile Appearance Design Ideation with Kansei Engineering and Generative AI, in: *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, 2024, pp. 1–19.
[32] D. Choi, S. Hong, J. Park, J.J.Y. Chung and J. Kim, CreativeConnect: Supporting Reference Recombination for Graphic Design Ideation with Generative AI, in: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–25.
[33] J.J.Y. Chung and M. Kreminski, Patchview: LLM-Powered Worldbuilding with Generative Dust and Magnet Visualization, in: *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, 2024, pp. 1–19.
[34] J.J.Y. Chung, W. Kim, K.M. Yoo, H. Lee, E. Adar and M. Chang, TaleBrush: Sketching stories with generative pretrained language models, in: *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 2022, pp. 1–19.
[35] F. Ciroku, J. de Berardinis, J. Kim, A. Meroño-Peñuela, V. Presutti and E. Simperl, RevOnt: Reverse engineering of competency questions from knowledge graphs via language models, *Journal of Web Semantics* (2024), 100822.
[36] J.H. Claessen and J.J. Van Wijk, Flexible linked axes for multivariate data visualization, *IEEE Transactions on Visualization and Computer Graphics* **17**(12) (2011), 2310–2316.
[37] N.J. Cooke, Knowledge elicitation, *Handbook of applied cognition* (1999), 479–509.
[38] N. Cross, Expertise in design: an overview, *Design studies* **25**(5) (2004), 427–441.
[39] J. de Berardinis, V.A. Carriero, N. Jain, N. Lazzari, A. Meroño-Peñuela, A. Poltronieri and V. Presutti, The polifonia ontology network: Building a semantic backbone for musical heritage, in: *International Semantic Web Conference*, Springer, 2023, pp. 302–322.
[40] E. De Bono and E. Zimbalist, *Lateral thinking*, Penguin London, 1970.
[41] A. De Nicola and M. Missikoff, A lightweight methodology for rapid ontology engineering, *Communications of the ACM* **59**(3) (2016), 79–86.
[42] Design Council, The Double Diamond: A universally accepted depiction of the design process, 2005, Accessed: April 28, 2025. https://www.designcouncil.org.uk/our-resources/the-double-diamond/.
[43] A. Dix, *Human-computer interaction*, Vol. 1, Pearson Education, 2004.
[44] S.P. Dow, A. Glassco, J. Kass, M. Schwarz, D.L. Schwartz and S.R. Klemmer, 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.
[45] N. Fathallah, A. Das, S.D. Giorgis, A. Poltronieri, P. Haase and L. Kovriguina, Neon-GPT: a large language model-powered pipeline for ontology learning, in: *European Semantic Web Conference*, Springer, 2024, pp. 36–50.
[46] J. Fereday and E. Muir-Cochrane, Demonstrating rigor using thematic analysis: A hybrid approach of inductive and deductive coding and theme development, *International journal of qualitative methods* **5**(1) (2006), 80–92.
[47] J.L. Fleiss and J. Cohen, The Equivalence of Weighted Kappa and the Intraclass Correlation Coefficient as Measures of Reliability, *Educational and Psychological Measurement* **33**(3) (1973), 613–619. doi:10.1177/001316447303300309.
[48] G.W. Furnas, Generalized fisheye views, *Acm Sigchi Bulletin* **17**(4) (1986), 16–23.
[49] M. Ghajargar, J. Bardzell and L. Lagerkvist, A redhead walks into a bar: experiences of writing fiction with artificial intelligence, in: *Proceedings of the 25th international academic MindTrek conference*, 2022, pp. 230–241.
[50] G. Gibbs, Analyzing Qualitative Data, *(No Title)* (2007).
[51] S. Greenberg, S. Carpendale, N. Marquardt and B. Buxton, *Sketching user experiences: The workbook*, Elsevier, 2012.
[52] M. Grüninger and M.S. Fox, The role of competency questions in enterprise engineering, in: *Benchmarking—Theory and practice*, Springer, 1995, pp. 22–31.
[53] J.P. Guilford, Three faces of intellect. (1961).
[54] Y. Guo, H. Shao, C. Liu, K. Xu and X. Yuan, Prompthis: Visualizing the process and influence of prompt editing during text-to-image creation, *IEEE Transactions on Visualization and Computer Graphics* (2024).
[55] K. Halskov and C. Lundqvist, Filtering and informing the design space: Towards design-space thinking, *ACM Transactions on Computer-Human Interaction (TOCHI)* **28**(1) (2021), 1–28.

[56] S.G. Hart and L.E. Staveland, Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research, in: *Advances in psychology*, Vol. 52, Elsevier, 1988, pp. 139–183.
[57] D. Hayatpur, D. Wigdor and H. Xia, Crosscode: Multi-level visualization of program execution, in: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023, pp. 1–13.
[58] C. Heape, The Design Space: the design process as the construction, exploration and expansion of a conceptual space (2007).
[59] M.N. Hoque, T. Mashiat, B. Ghai, C.D. Shelton, F. Chevalier, K. Kraus and N. Elmqvist, The HaLLMark Effect: Supporting Provenance and Transparent Use of Large Language Models in Writing with Interactive Visualization, in: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–15.
[60] R. Huang, H. Lin, C. Chen, K. Zhang and W. Zeng, Plantography: Incorporating iterative design process into generative artificial intelligence for landscape rendering, in: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–19.
[61] M. Huh, Y.-H. Peng and A. Pavel, GenAssist: Making image generation accessible, in: *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023, pp. 1–17.
[62] R.J. Jacob, New human-computer interaction techniques, *NATO ASI SERIES F COMPUTER AND SYSTEMS SCIENCES* **129** (1994), 131–131.
[63] D.G. Jansson and S.M. Smith, Design fixation, *Design studies* **12**(1) (1991), 3–11.
[64] N. Jog and B. Shneiderman, Information visualization with smooth zooming on an starfield display, in: *Proc. IFIP Conf. Visual Databases*, Vol. 3, 1995, pp. 1–10.
[65] C.M. Keet, Z. Mahlaza and M.-J. Antia, CLaRO: a controlled language for authoring competency questions, in: *Research Conference on Metadata and Semantics Research*, Springer, 2019, pp. 3–15.
[66] E.A. Kemp, The role of the individual project in teaching knowledge acquisition, in: *Proceedings 1996 International Conference Software Engineering: Education and Practice*, IEEE, 1996, pp. 138–143.
[67] E.F. Kendall and D.L. McGuinness, *Ontology engineering*, Morgan & Claypool Publishers, 2019.
[68] J. Kim, S. Suh, L.B. Chilton and H. Xia, Metaphorian: Leveraging large language models to support extended metaphor creation for science writing, in: *Proceedings of the 2023 ACM Designing Interactive Systems Conference*, 2023, pp. 115–135.
[69] T.S. Kim, Y. Lee, M. Chang and J. Kim, Cells, generators, and lenses: Design framework for object-oriented interaction with large language models, in: *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023, pp. 1–18.
[70] T. Kuhn, A Survey and Classification of Controlled Natural Languages, *Computational Linguistics* **40**(1) (2014), 121–170. doi:10.1162/COLI_a_00168. https://aclanthology.org/J14-1005/.
[71] T. Kuhn, A survey and classification of controlled natural languages, *Computational linguistics* **40**(1) (2014), 121–170.
[72] M. Lee, P. Liang and Q. Yang, Coauthor: Designing a human-ai collaborative writing dataset for exploring language model capabilities, in: *Proceedings of the 2022 CHI conference on human factors in computing systems*, 2022, pp. 1–19.
[73] S. Lin, J. Warner, J. Zamfirescu-Pereira, M.G. Lee, S. Jain, S. Cai, P. Lertvittayakumjorn, M.X. Huang, S. Zhai, B. Hartmann et al., Rambler: Supporting Writing With Speech via LLM-Assisted Gist Manipulation, in: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–19.
[74] V. Liu, Beyond text-to-image: Multimodal prompts to explore generative ai, in: *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023, pp. 1–6.
[75] V. Liu, J. Vermeulen, G. Fitzmaurice and J. Matejka, 3DALL-E: Integrating text-to-image AI in 3D design workflows, in: *Proceedings of the 2023 ACM designing interactive systems conference*, 2023, pp. 1955–1977.
[76] J.D. Lomas, M. Karac and M. Gielen, Design space cards: using a card deck to navigate the design space of interactive play, *Proceedings of the ACM on Human-Computer Interaction* **5**(CHI PLAY) (2021), 1–21.
[77] L. Long, C. Xinyi, W. Ruoyu, L. Toby Jia-Jun and L. Ray, Sketchar: Supporting character design and illustration prototyping using generative AI, *Proceedings of the ACM on Human-Computer Interaction* **8**(CHI PLAY) (2024), 337.
[78] A. MacEachren, D. Xiping, F. Hardisty, D. Guo and G. Lengerich, Exploring high-D spaces with multiform matrices and small multiples, in: *IEEE Symposium on Information Visualization 2003 (IEEE Cat. No. 03TH8714)*, IEEE, 2003, pp. 31–38.
[79] D. Masson, S. Malacria, G. Casiez and D. Vogel, Directgpt: A direct manipulation interface to interact with large language models, in: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–16.
[80] D. McKerlie and A. MacLean, Reasoning with design rationale: practical experience with design space analysis, *Design Studies* **15**(2) (1994), 214–226.
[81] G.A. Miller, WordNet: a lexical database for English, *Communications of the ACM* **38**(11) (1995), 39–41.
[82] G. Miriam Jacob, S. Sahu, C. Ito, M. Kawamura and B. Stenger, Text2Illustration: Sampling and Composing Creatives With Text, in: *Proceedings of the 7th Joint International Conference on Data Science & Management of Data (11th ACM IKDD CODS and 29th COMAD)*, 2024, pp. 538–541.
[83] P. Mirowski, K.W. Mathewson, J. Pittman and R. Evans, Co-writing screenplays and theatre scripts with language models: Evaluation by industry professionals, in: *Proceedings of the 2023 CHI conference on human factors in computing systems*, 2023, pp. 1–34.
[84] D. NORMAN, Some observations on mental models, *Mental Models* (1983), 7–14.
[85] D.A. Norman, *The design of everyday things*, Basic Books, 2002.
[86] N.F. Noy, Ontology Development 101: A Guide to Creating Your First Ontology, Technical Report, KSL-01-05, SMI-2001-0880, Stanford University, Knowledge Systems Laboratory, 2001. https://protege.stanford.edu/publications/ontology_development/ontology101.pdf.
[87] H. Osone, J.-L. Lu and Y. Ochiai, BunCho: ai supported story co-creation via unsupervised multitask learning to increase writers' creativity in japanese, in: *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–10.

[88] M.J. Page, J.E. McKenzie, P.M. Bossuyt, I. Boutron, T.C. Hoffmann, C.D. Mulrow, L. Shamseer, J.M. Tetzlaff, E.A. Akl, S.E. Brennan et al., The PRISMA 2020 statement: an updated guideline for reporting systematic reviews, *bmj* **372** (2021).
[89] M. Poveda-Villalón, A. Fernández-Izquierdo, M. Fernández-López and R. García-Castro, LOT: An industrial oriented ontology engineering framework, *Engineering Applications of Artificial Intelligence* **111** (2022), 104755.
[90] V. Presutti, E. Daga, A. Gangemi and E. Blomqvist, eXtreme design with content ontology design patterns, in: *Proc. Workshop on Ontology Patterns*, 2009, pp. 83–97.
[91] H.X. Qin, S. Jin, Z. Gao, M. Fan and P. Hui, CharacterMeet: Supporting Creative Writers' Entire Story Character Construction Processes Through Conversation with LLM-Powered Chatbot Avatars, in: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–19.
[92] Y. Ren, A. Parvizi, C. Mellish, J.Z. Pan, K. Van Deemter and R. Stevens, Towards competency question-driven ontology authoring, in: *The Semantic Web: Trends and Challenges: 11th International Conference, ESWC 2014, Anissaras, Crete, Greece, May 25-29, 2014. Proceedings 11*, Springer, 2014, pp. 752–767.
[93] M. Reza, N.M. Laundry, I. Musabirov, P. Dushniku, Z.Y.M. Yu, K. Mittal, T. Grossman, M. Liut, A. Kuzminykh and J.J. Williams, ABScribe: Rapid Exploration & Organization of Multiple Writing Variations in Human-AI Co-Writing Tasks using Large Language Models, in: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–18.
[94] G.W. Ryan, H.R. Bernard et al., Data management and analysis methods, *Handbook of qualitative research* **2**(1) (2000), 769–802.
[95] M. Sarkar and M.H. Brown, Graphical fisheye views, *Communications of the ACM* **37**(12) (1994), 73–83.
[96] F. Sclano, P. Velardi et al., TermExtractor: A Web application to learn the common terminology of interest groups and research communities, in: *Proceedings of the 9th Conference on Terminology and Artificial Intelligence*, Citeseer, 2007, pp. 1–10.
[97] N. Shadbolt and A.M. Burton, The empirical study of knowledge elicitation techniques, *ACM SIGART Bulletin* (1989), 15–18.
[98] M. Shaw, The role of design spaces, *IEEE software* **29**(1) (2011), 46–50.
[99] B. SHNEIDERMAN, The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations, *VL96* (1996), 336–343.
[100] N. Singh, G. Bernal, D. Savchenko and E.L. Glassman, Where to hide a stolen elephant: Leaps in creative writing with multimodal machine intelligence, *ACM Transactions on Computer-Human Interaction* **30**(5) (2023), 1–57.
[101] K. Son, D. Choi, T.S. Kim, Y.-H. Kim and J. Kim, Genquery: Supporting expressive visual search with generative models, in: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–19.
[102] M.C. Suárez-Figueroa, A. Gómez-Pérez and M. Fernandez-Lopez, The NeOn Methodology framework: A scenario-based methodology for ontology development, *Applied ontology* **10**(2) (2015), 107–145.
[103] S. Suh, B. Min, S. Palani and H. Xia, Sensecape: Enabling multilevel exploration and sensemaking with large language models, in: *Proceedings of the 36th annual ACM symposium on user interface software and technology*, 2023, pp. 1–18.
[104] S. Suh, M. Chen, B. Min, T.J.-J. Li and H. Xia, Luminate: Structured generation and exploration of design space with large language models for human-ai co-creation, in: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–26.
[105] N. Sultanum and A. Srinivasan, Datatales: Investigating the use of large language models for authoring data-driven articles, in: *2023 IEEE Visualization and Visual Analytics (VIS)*, IEEE, 2023, pp. 231–235.
[106] Y. Sun, Z. Li, K. Fang, C.H. Lee and A. Asadipour, Language as reality: a co-creative storytelling game experience in 1001 nights using generative AI, in: *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, Vol. 19, 2023, pp. 425–434.
[107] B. Swanson, K. Mathewson, B. Pietrzak, S. Chen and M. Dinalescu, Story centaur: Large language model few shot learning as a creative writing tool, in: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, 2021, pp. 244–256.
[108] A. Taheri, M. Izadi, G. Shriram, N. Rostamzadeh and S. Kane, Breaking barriers to creative expression: Co-designing and implementing an accessible text-to-image interface, *arXiv preprint arXiv:2309.02402* (2023).
[109] D.R. Thomas, A general inductive approach for analyzing qualitative evaluation data, *American journal of evaluation* **27**(2) (2006), 237–246.
[110] J. Tidwell, *Designing interfaces: Patterns for effective interaction design*, " O'Reilly Media, Inc.", 2010.
[111] B. Tilekbay, S. Yang, M.A. Lewkowicz, A. Suryapranata and J. Kim, ExpressEdit: Video Editing with Natural Language and Sketching, in: *Proceedings of the 29th International Conference on Intelligent User Interfaces*, 2024, pp. 515–536.
[112] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser and I. Polosukhin, Attention is all you need, *Advances in neural information processing systems* **30** (2017).
[113] Q. Wan, X. Feng, Y. Bei, Z. Gao and Z. Lu, Metamorpheus: Interactive, Affective, and Creative Dream Narration Through Metaphorical Visual Storytelling, in: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–16.
[114] B. Wang, Y. Li, Z. Lv, H. Xia, Y. Xu and R. Sodhi, LAVE: LLM-powered agent assistance and language augmentation for video editing, in: *Proceedings of the 29th International Conference on Intelligent User Interfaces*, 2024, pp. 699–714.
[115] S. Wang, S. Petridis, T. Kwon, X. Ma and L.B. Chilton, PopBlends: Strategies for conceptual blending with large language models, in: *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023, pp. 1–19.
[116] S. Wang, Z. Ning, A. Truong, M. Dontcheva, D. Li and L.B. Chilton, PodReels: Human-AI Co-Creation of Video Podcast Teasers, in: *Proceedings of the 2024 ACM Designing Interactive Systems Conference*, 2024, pp. 958–974.
[117] S. Wang, S. Menon, T. Long, K. Henderson, D. Li, K. Crowston, M. Hansen, J.V. Nickerson and L.B. Chilton, ReelFramer: Human-AI co-creation for news-to-video translation, in: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–20.

[118] Z. Wang, Y. Huang, D. Song, L. Ma and T. Zhang, Promptcharm: Text-to-image generation through multi-modal prompting and refinement, in: *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–21.
[119] C. Ware and S. Osborne, Exploration and virtual camera control in virtual three dimensional environments, in: *Proceedings of the 1990 symposium on Interactive 3D graphics*, 1990, pp. 175–183.
[120] B. Westerlund, Design space conceptual tool–grasping the design process, *Nordes* (2005).
[121] M. Wiberg and E. Stolterman, What makes a prototype novel? A knowledge contribution concern for interaction design research, in: *Proceedings of the 8th Nordic conference on human-computer interaction: fun, fast, foundational*, 2014, pp. 531–540.
[122] R.H. Willemsen, I.C. de Vink, E.H. Kroesbergen and A.W. Lazonder, The role of creative thinking in children's scientific reasoning, *Thinking Skills and Creativity* **49** (2023), 101375.
[123] C. Williamson and B. Shneiderman, The Dynamic HomeFinder: Evaluating dynamic queries in a real-estate information exploration system, in: *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, 1992, pp. 338–346.
[124] D. Wiśniewski, J. Potoniec, A. Ławrynowicz and C.M. Keet, Analysis of ontology competency questions and their formalizations in SPARQL-OWL, *Journal of Web Semantics* **59** (2019), 100534.
[125] L. Xie, C. Zheng, H. Xia, H. Qu and C. Zhu-Tian, Waitgpt: Monitoring and steering conversational llm agent in data analysis with on-the-fly code visualization, in: *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, 2024, pp. 1–14.
[126] H. Yakura and M. Goto, IteraTTA: An interface for exploring both text prompts and audio priors in generating music with text-to-audio models, *arXiv preprint arXiv:2307.13005* (2023).
[127] Z. Yan, C. Yang, Q. Liang and X. Chen, XCreation: A graph-based crossmodal generative creativity support tool, in: *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023, pp. 1–15.
[128] D. Yang, E. Kleinman, G.M. Troiano, E. Tochilnikova and C. Harteveld, Snake Story: Exploring Game Mechanics for Mixed-initiative Co-creative Storytelling Games, in: *Proceedings of the 19th International Conference on the Foundations of Digital Games*, 2024, pp. 1–11.
[129] J.S. Yi, Y. ah Kang, J. Stasko and J.A. Jacko, Toward a deeper understanding of the role of interaction in information visualization, *IEEE transactions on visualization and computer graphics* **13**(6) (2007), 1224–1231.
[130] R.J. Youmans and T. Arciszewski, Design fixation: Classifications and modern methods of prevention, *AI EDAM* **28**(2) (2014), 129–137.
[131] A. Yuan, A. Coenen, E. Reif and D. Ippolito, Wordcraft: story writing with large language models, in: *Proceedings of the 27th International Conference on Intelligent User Interfaces*, 2022, pp. 841–852.
[132] B. Zhang, V.A. Carriero, K. Schreiberhuber, S. Tsaneva, L.S. González, J. Kim and J. de Berardinis, OntoChat: a framework for conversational ontology engineering using language models, in: *European Semantic Web Conference*, Springer, 2024, pp. 102–121.
[133] Z. Zhang, J. Gao, R.S. Dhaliwal and T.J.-J. Li, Visar: A human-ai argumentative writing assistant with visual programming and rapid draft prototyping, in: *Proceedings of the 36th annual ACM symposium on user interface software and technology*, 2023, pp. 1–30.
[134] Y. Zhao, B. Zhang, X. Hu, S. Ouyang, J. Kim, N. Jain, J. de Berardinis, A. Meroño-Peñuela and E. Simperl, Improving Ontology Requirements Engineering with OntoChat and Participatory Prompting, in: *Proceedings of the AAAI Symposium Series*, Vol. 4, 2024, pp. 253–257.
[135] Y. Zhao, Z. Li, Y. Wang, X. Cai, X. Zhou, Y. Yan, K. Jin, S. Ding, Y. Shao, J. Cao et al., DreamDirector: Designing a Generative AI System to Aid Therapists in Treating Clients' Nightmares, in: *Proceedings of the 30th International Conference on Intelligent User Interfaces*, 2025, pp. 553–578.
[136] A. Zhu, L. Martin, A. Head and C. Callison-Burch, CALYPSO: LLMs as Dungeon Master's Assistants, in: *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, Vol. 19, 2023, pp. 380–390.