

OntoScope: Using a Divergent-Convergent Interaction Framework to Support LLM-based Ontology Scoping

Yihang Zhao
yihang.zhao@kcl.ac.uk
King's College London
London, United Kingdom

Albert Meroño Peñuela
albert.merono@kcl.ac.uk
King's College London
London, United Kingdom

Elena Simperl
elena.simperl@kcl.ac.uk
King's College London
London, United Kingdom
Technical University of Munich
Munich, Germany

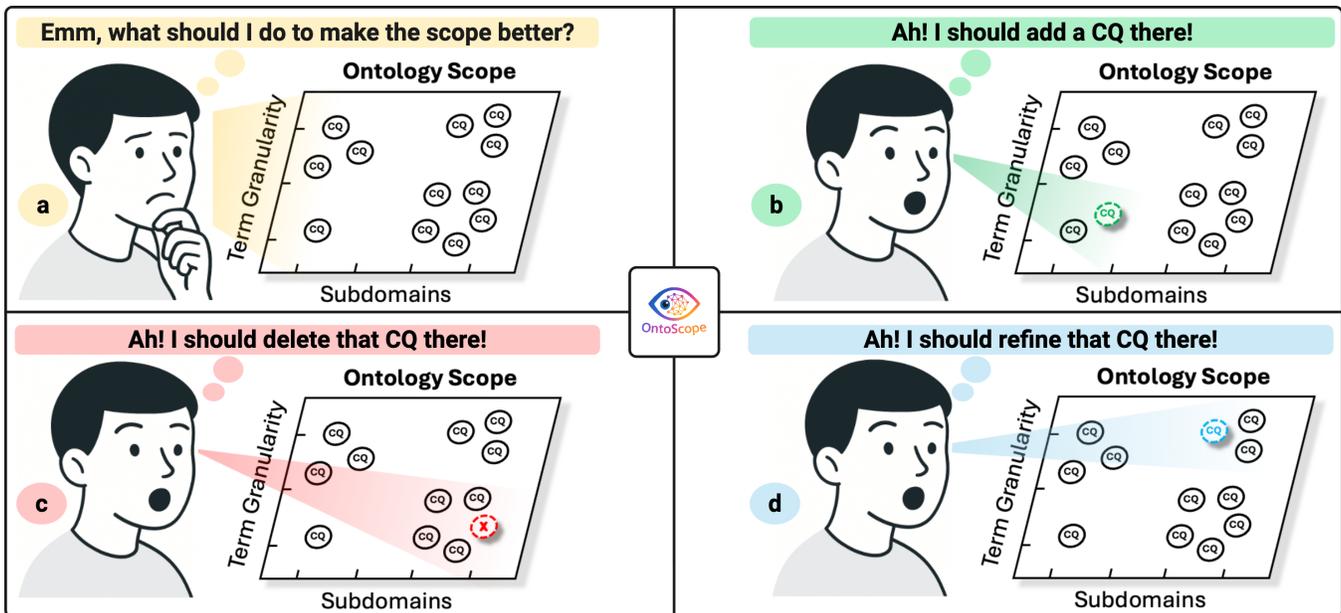


Figure 1: A conceptual summary of OntoScope’s ontology scoping workflow. The ontology scope is modelled as a 2D grid of subdomains (x-axis) and term granularities (y-axis). Each cell is a conceptual region, defined by a specific subdomain and granularity level, where competency questions (CQs) are placed and assessed. CQs are functional requirements that define the domain knowledge (concepts and relations) an ontology must model. To audit ontology scope, OntoScope will help ontology engineers (a) identify gaps and overlaps from a global perspective; (b) use *divergent thinking* to add CQs in selected cells to cover under-represented regions, enrich CQs with terminologies, or extend dimensions by adding axis values, with the LLM suggesting additional CQs, terms, or axis values; (c) use *convergent thinking* to delete similar or irrelevant CQs, terms, and axis values within and across cells, while the LLM adapts to avoid repeating similar suggestions; and (d) continue *convergent thinking* to refine CQs, terms, and axis values for greater precision, using LLM-suggested alternatives and rephrasings. This iterative divergent–convergent process lets ontology engineers and the LLM progressively and systematically define the ontology scope.

Abstract

An ontology is a formal, explicit specification of a shared conceptualization that, with problem-solving and reasoning methods, supports efficient semantic technology development. In ontology engineering, Competency Questions (CQs) capture functional requirements that define an ontology’s application domain. Auditing this domain scope with CQs is challenging because in nature, there are no clear domain boundaries, and ontology engineers must then decide which subdomains to cover (horizontal coverage) and how much detail to model (vertical granularity) in an ontology. LLM-based systems can generate many candidate CQs to guide these decisions, but current tools underuse this potential: they lack support for users’ divergent (lateral) and convergent (vertical) thinking in a visualized CQs space organized by coverage and granularity. As a result, users struggle to systematically decide which CQs to adopt, discard, or refine. We propose an interaction framework that fills

this gap, demonstrated through OntoScope, an LLM-based interactive system, and a user study with 15 ontology engineers. To our knowledge, this is the first validated interaction framework with an LLM-based system that helps ontology engineers audit domain boundaries and unifies fragmented, expert-driven ontology scoping practices into a coherent, accessible approach. More broadly, it shows how LLM-based systems can transparently and accountably support a wider range of knowledge-intensive tasks.

CCS Concepts

• Human-centered computing; • Information systems;

Keywords

Knowledge works, Knowledge engineering, Ontology engineering, KE, KG, OE, Requirements engineering, RE, Competency questions, CQ, Generative AI, GenAI, Artificial intelligence, AI, Large language

models, LLM, User interface, UI, Interaction technics, Interaction design, IxD, Computational creativity, Creative thinking, Divergent thinking, Convergent thinking, Cognitive process

ACM Reference Format:

Yihang Zhao, Albert Meroño Peñuela, and Elena Simperl. 2018. OntoScope: Using a Divergent-Convergent Interaction Framework to Support LLM-based Ontology Scoping. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 18 pages. <https://doi.org/XXXXXXX.XXXXXXX>

1 Introduction

In philosophy, an ontology is a categorical structure that systematizes a conception of reality [57]. In information and communication technologies (ICT), it provide formal, explicit, reusable pieces of declarative knowledge. Combined with problem-solving methods and inference, it can be used to support semantic technologies and applications such as information retrieval, information extraction, and data or process integration, while reducing development effort and cost [57, 90]. OE studies the activities needed to construct and maintain ontologies, and the related methodologies, tools, and representation languages [53].

CQs are essential in most OE methodologies [1, 72, 96–98]. They express the ontology’s functional requirements as natural language questions and support the elicitation of candidate terminology (e.g., concepts and relations) that defines the intended knowledge scope [92, 97, 98, 105, 112]. Beyond requirements elicitation, CQs also support later OE activities, including ontology validation [18, 73], consumption [43], and reuse [5, 11]. Their concrete form varies with the underlying methodology [56, 92], question patterns [127], templates [9, 74], archetypes [102], intended use [8], and level of automation [6, 101]. In this study, we use the CQ patterns from the LLM-based CQs generation framework RevOnt [29], evaluated on 20 representative domains at Wikidata¹ (the largest collaboratively curated knowledge graph, covering widely shared knowledge across many domains [122]). RevOnt [29] specifies three CQ patterns (Fig. 2), each differing in which element of an abstract Resource Description Framework (RDF) triple is treated as the answer. An RDF triple², written as (subject, predicate/property, object), encodes a single fact in a knowledge graph; abstracting such triples yields schema-level statements suitable for ontology specification.

With the emergence of LLMs, several semi- or fully automated, empirically validated CQs generation approaches, such as RevOnt [29], RETROFIT-CQs [6], AgOCQs [10], OntoChat [130, 138], and related work [7, 101], have been developed to explore LLM-based support for requirements elicitation in OE. These approaches typically generate CQs from verbalized triples derived from existing knowledge resources, including ontologies [6, 7], knowledge graphs [29], specialized corpora [10], or ontology user stories [130, 138]. They are usually evaluated on benchmarks drawn from the same resources, e.g., corresponding CQs published alongside each ontology [6, 7, 101], or via expert judgments [10, 29]. Thus, existing work primarily evaluates how well LLMs generate CQs from knowledge bases, rather than designing interactive systems that help ontology

engineers navigate and select proposed CQs to define the target ontology scope. an equally crucial requirements elicitation activity in OE methodologies [4, 37, 87].

Our study builds on the UPON Lite methodology [37], derived from the UPON framework [38] and, ultimately, the Unified Software Development Process [68]. In UPON Lite [37], ontology scoping is the first development step and produces a domain-specific lexicon that supports subsequent modelling. Constructing this lexicon involves decisions along two key dimensions: a horizontal (coverage) dimension, specifying how many subdomains the candidate CQs address, and a vertical (granularity) dimension, specifying the level of detail in the term representation [37]. To support these decisions, ontology engineers commonly use tools such as Protégé [89], TopBraid EDG [117], ontoKEM [116], and the PoolParty Semantic Suite [51]. These tools typically offer list-based interfaces for managing candidate terminologies and tracking ontology scope. However, users must navigate large, nested hierarchies to locate and edit individuals, which impedes maintaining a global overview and hinders reasoning about the intended scope [87]. Consequently, ontology engineers often resort to external tools, such as mind-mapping software (e.g., FreeMind [88]) or collaborative whiteboards (e.g., Miro [86]), to arrange CQs visually for ontology scope auditing. Yet these tools lack dedicated generative support for systematically expanding or refining the ontology scope (Miro’s generative features are not specialised for CQs generation or refinement), and there is no empirical evidence that such user-defined visualizations improve ontology scope auditing decisions.

LLM-based systems can automatically generate large sets of candidate CQs with proposed terminologies to help ontology engineers delimit ontology scope [6, 7, 10, 29, 101, 130, 138]. Yet current LLM-based CQs generation is not fully used to audit domain boundaries. It does not support users’ divergent (lateral) and convergent (vertical) thinking over a visual landscape of CQs organized by ontology coverage and granularity. As a result, ontology engineers lack systematic support for deciding which CQs to include, discard, or refine when iteratively shaping ontology scope. Divergent and convergent thinking processes [31, 36, 39, 52, 58, 67, 125, 132], combined with an appropriate interaction framework [22, 54, 115], that is, interaction techniques and interfaces grounded in HCI and data visualization [65, 108], have been widely validated in creativity-support tools to enable designers’ fluid exploration, evaluation, and refinement of ideas toward a desired outcome [77, 113, 114]. We therefore pose the following research questions (RQs) to investigate how these approaches can transfer to ontology scoping:

- **RQ1:** What interaction framework is needed in LLM-based interactive systems to support ontology engineers in ontology scoping?
- **RQ2:** How can this interaction framework be developed into an LLM-based interactive system to support ontology engineers in ontology scoping?
- **RQ3:** How useful is this interaction framework for supporting ontology engineers in defining the scope of an ontology?

To address these RQs, **we made the following contributions:**

¹https://www.wikidata.org/wiki/Wikidata:Main_Page

²<https://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>

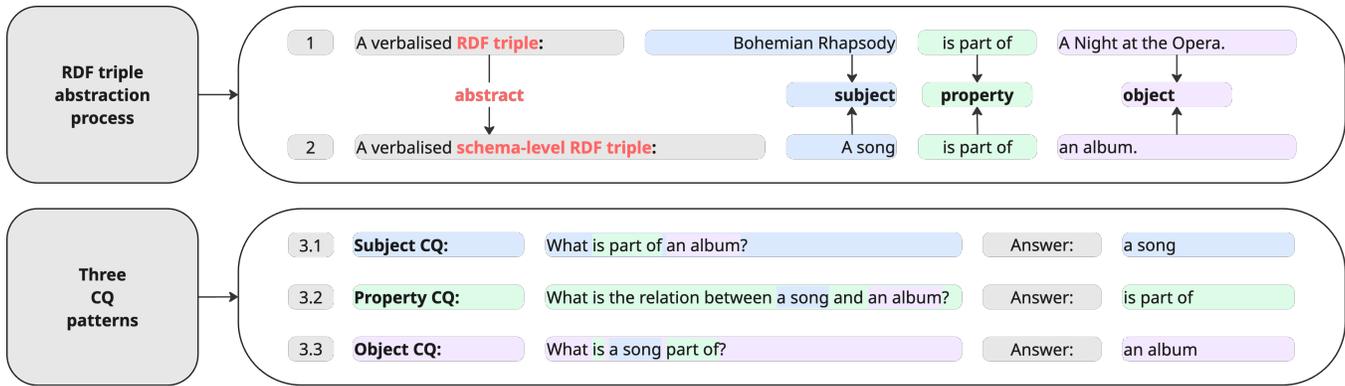


Figure 2: Example of constructing three CQ patterns in RevOnt [29]. Step 1 verbalises an RDF triple from the music domain. Step 2 abstracts it into a schema-level triple for ontology definition. Steps 3.1–3.3 derive three CQ patterns from this schema-level triple, each determined by which part of the triple (subject, property/predicate, object) is treated as the answer.

- Developed **two taxonomies**³ via a literature search, categorizing interaction techniques (4 categories, 7 subcategories) and user interfaces (UIs) (9 categories, 14 subcategories) in LLM-based systems for divergent and convergent thinking in creative domains.
- Designed **an interaction framework** (Table 2) through a design thinking workshop, using these taxonomies to support ontology engineers in defining ontology scope.
- Implemented **OntoScope**⁴, an LLM-based interactive system that applies this framework to ontology scoping.
- Conducted **a user and technical evaluation** (Section 5) to assess how well the framework supports ontology scoping through OntoScope.

These contributions advance OE practice by: (1) OntoScope, the first LLM-based system that unifies fragmented ontology scoping workflows; (2) a divergent–convergent design framework that empirically supports coverage \times granularity reasoning for rigorous scope auditing; and (3) the first empirical evidence that an LLM-based divergent–convergent interaction framework measurably improves ontology scoping. This interaction framework also (1) generalizes to other knowledge-intensive scoping tasks that balance breadth and depth and (2) shows how iterative exploration and refinement of generative outputs in a structured visual workspace supports transparent, accountable decision-making in knowledge-intensive tasks.

2 Related work

We present prior research and concepts relevant to our work: ontology scoping with CQs in Section 2.1, divergent and convergent thinking in Section 2.2, and support for these thinking processes in Section 2.3.

2.1 Ontology scoping with CQs

In OE, ontology scoping with CQs defines what an ontology should represent using natural language questions and expected answers.

These questions specify the knowledge an ontology or ontology-based application must answer correctly [55, 118], thereby measuring ontology scope [76, 100]. The term “Competency Question” comes from human resources [110], where predefined questions with known answers assess candidates’ suitability. To make CQs a reliable scope measure, suitable requirements elicitation and auditing techniques must ensure a complete, representative CQs set [100].

CQs are typically elicited from domain experts and other stakeholders, such as the intended users of ontology based systems, through unstructured interviews and workshops in early stages, when ontology engineers are still unfamiliar with the scope [30, 35, 100, 130]. These methods often reflect ontology engineers’ prior assumptions about what the ontology should include, constraining responses and limiting CQs coverage [91, 130, 135]. This interviewer framing effect is well documented in HCI [30, 40, 41]. Once ontology engineers better understand the domain, structured techniques such as card sorting [106], triad analysis [103], and the twenty questions method [75] can classify and validate CQs-elicited terms, further refining ontology scope. However, these manual processes are time-intensive for large OE projects.

To address this limitation, Natural language processing (NLP) techniques such as named entity recognition, information extraction, and lexical–syntactic analysis have been used to extract terminologies from domain corpora [61, 63, 80, 131, 137] and automatically fill predefined CQ templates [9, 19, 74, 78, 102, 127], thereby generating CQs for scoping the target ontology. Building on these methods, large pre-trained models have been used to capture broader semantic relationships and generate richer content [6, 7, 10, 29, 101]. However, most work evaluates CQs generation from existing knowledge sources rather than how generated CQs help ontology engineers audit the target ontology’s scope. A notable exception is *OntoChat* [130, 134, 135, 138], an LLM-based conversational agent that simulates ontology engineers and supports iterative elicitation and refinement of user stories with domain experts, which are then converted into CQs. Yet preliminary user studies show that its linear conversational interface tends to follow paths shaped by users’ initial inputs [34, 42, 69, 113, 128], limiting

³https://github.com/King-s-Knowledge-Graph-Lab/OntoScope/blob/main/attached_assets/Codebook.xlsx

⁴<https://github.com/King-s-Knowledge-Graph-Lab/OntoScope>

exploration of alternative stories and yielding CQs that may not fully capture the intended ontology scope.

Other tools used to audit candidate CQs and terminologies for ontology scoping, though with limited success [87], include Protégé [89], TopBraid EDG [117], ontoKEM [116], and PoolParty Semantic Suite [51]. Their list-based interfaces force users to scroll through long, nested hierarchies, hindering overview and reasoning about ontology scope [87]. Mind-mapping tools such as FreeMind [88] and collaborative platforms like Miro [86] have also been used, but they were not designed or empirically validated for ontology scoping. Thus, effective support for ontology engineers remains unclear, a gap this study addresses.

2.2 Divergent and convergent thinking

Supporting ontology engineers in auditing domain boundaries means helping them select, discard, or refine CQs to define ontology scope [35, 37]. This process parallels divergent and convergent thinking [31, 39, 52, 67, 132], as in Poincaré’s four-phase model [21, 48, 82]. Divergent thinking dominates the “Preparation” and “Incubation” phases, when many ideas are generated [12, 50], while convergent thinking drives the “Insight” and “Revision” phases, which narrow options, select promising ideas, and refine them into coherent solutions [33, 109]. Creativity thus requires both idea generation and subsequent selection and refinement [62, 139].

Balancing divergent and convergent thinking helps avoid “fixation” [34, 42, 113, 114, 128], where people converge too quickly on a sub-optimal idea and over-refine it instead of exploring alternatives. Recent LLM-based CQs generation tools mainly support prompt refinement based on system output, which can foster fixation by prioritising refinement over exploration [130, 134, 138]. Text-to-image systems likewise generate multiple images but still emphasise prompt refinement toward a single preferred one, again risking fixation [23, 45]. To avoid fixation in ontology requirements elicitation, requirements exploration (creating diverse CQs to probe domain boundaries) must be coordinated with requirements refinement (evaluating and narrowing these CQs to an optimal subset) to define the intended ontology scope.

2.3 Supporting divergent and convergent thinking

To support divergent and convergent thinking, designers use a structured design space [15, 20, 123]. A design space is a conceptual “space of ideas organized by dimensions,” where ideas are designs, concepts, or solutions [59, 85]. It is often depicted as a Cartesian space, with variations as points defined by two dimensions [15, 59, 83, 107]. Brainstorming tools such as Miro⁵ use this format: users put ideas on sticky notes and position them along meaningful dimensions [59, 85]. Although Miro does not require a 2D layout, teams often assign meanings to axes and populate the grid with sticky notes as value combinations. Externalizing the design space makes variations, dimensions, and values explicit, enabling dimensional reasoning [7, 15]. This helps designers step back from individual ideas to view the entire space, compare ideas, find gaps, classify new artifacts, and see what makes them distinct or novel [113, 114, 124]. In our work, dimensional reasoning helps

⁵<http://miro.com/>

ontology engineers reflect on the domain coverage and granularity of terminology suggested by candidate CQs, so they can assess completeness, find gaps, and refine the target ontology’s scope.

Many LLM-based systems support divergent and convergent thinking in an externalized idea space, serving as creativity support tools [2, 64, 99, 113, 114]. They treat LLMs as enhancers of human creativity, helping users explore and navigate possibilities. Yet we lack a systematic understanding of the information visualization [17] and HCI [113] techniques these systems use to support such thinking in art and creativity. To address this, we draw on Shneiderman’s “Visual Information-Seeking Mantra” [17] and organize these techniques into three stages. In the “Overview” stage [14, 47, 104], UIs such as multidimensional spaces, hierarchical trees, networks, and timelines can be used to externalize a high-level design space, while interaction techniques such as spatial navigation, panning, and zooming out can be used to support broad exploration. In the “Zoom and Filter” stage [3, 16, 71, 126], UIs such as scroll bars, sliders, search fields, and checkboxes, combined with interaction techniques such as zooming in, sliding, querying, and faceted selection, can be used to help users focus on specific dimensions or value ranges while staying aware of the overall space. In the “Details-on-Demand” stage [17], UIs such as pop-ups, expandable panels, and clickable or hoverable points can be used to let users inspect detailed information about individual artifacts for comparison, validation, and decision-making without overload. We present taxonomies of these interaction techniques and UIs in Section 3.

3 An interaction framework for ontology scoping with CQs

We developed an interaction framework for LLM-based systems that support ontology engineers in ontology scoping through a two-part process. First, a literature search (Section 3.1) produced two taxonomies of empirically grounded interaction techniques and UIs for creative and artistic LLM-based systems. These taxonomies inform RQ1 by identifying validated interaction patterns for supporting divergent and convergent thinking, treated here as cognitive analogues to ontology scoping. Second, a design thinking workshop (Section 3.2) applied these taxonomies to develop the interaction framework for supporting ontology scoping. Section 3.3 then presents OntoScope’s design goals, illustrating this interaction framework in practice.

3.1 Literature search

We conducted a literature search using selected PRISMA 2020 elements [95], adapted to our research needs. Given the exploratory, qualitative nature of this work and the need for iterative methodological adaptation, several PRISMA checklist components (e.g., meta-analysis, risk-of-bias assessment), the standard PRISMA flow diagram, and prospective registration were not applicable. The full procedure is shown in Fig. 3 and the full methodological details are in Appendix A).

We applied the search query shown in Figure 4 in the ACM Digital Library, IEEE Xplore, and Scopus to identify papers published from 2017 onward (following the born of Transformer architecture [120]). We applied venue-level filtering following Hu et al. [65], yielding 3,275 papers were retained for screening using eight

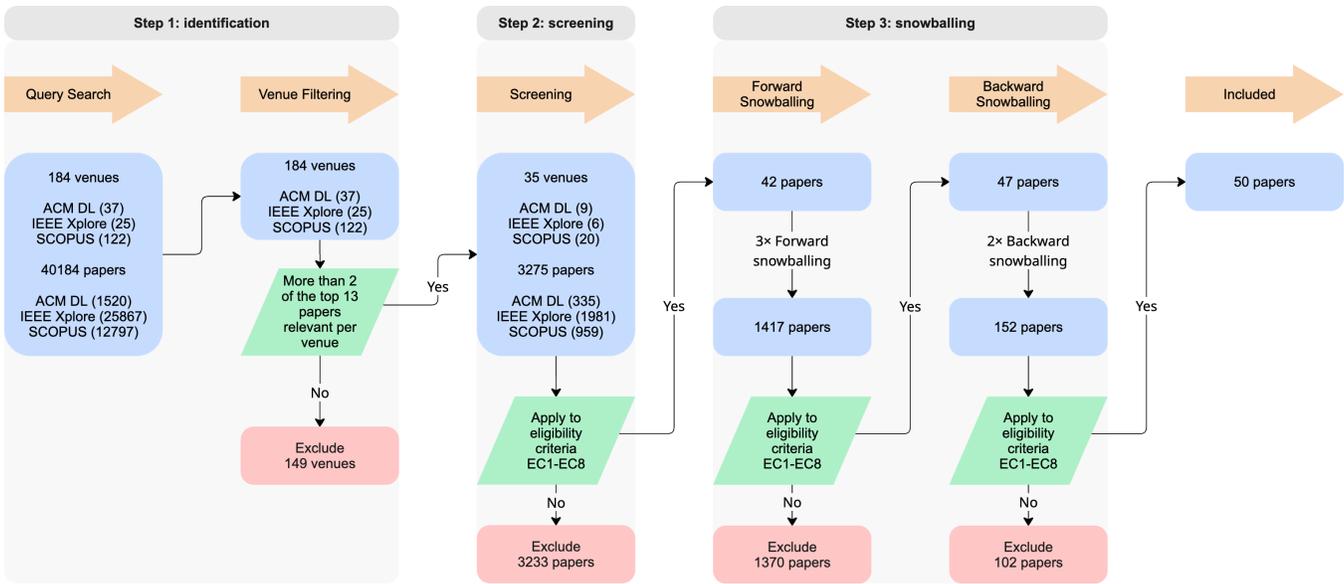


Figure 3: Literature search process. A *Query Search* across 184 venues in ACM Digital Library, IEEE Xplore, and Scopus yielded 40184 papers. *Venue Filtering* narrowed this to 35 venues and 3275 papers. Applying eligibility criteria (EC1–EC8) during *Screening* yielded 42 papers. Three rounds of *Forward Snowballing* added 1375 papers, of which 5 met the criteria. Two rounds of *Backward Snowballing* added 105 papers, of which 3 met the criteria. In total, 50 papers were included.

eligibility criteria (Table 1). We excluded secondary sources to reduce interpretive bias, but manually checked their references. We included systems with interaction techniques and UIs that were empirically evaluated for creative work.



Figure 4: Final search query used for the literature search.

Table 1: Exclusion criteria (EC) applied in the study.

EC-#	Description
EC-1	Full text not in English.
EC-2	Retracted publications.
EC-3	Non-peer-reviewed work.
EC-4	Secondary sources (e.g., surveys, reviews).
EC-5	Studies that do not present an LLM-based system supporting creative work in art or design.
EC-6	Systems without an interaction technique supporting the creative process.
EC-7	Systems without a UI supporting the creative process.
EC-8	Papers that do not evaluate how the interaction technique and/or UI support the creative process.

Most exclusions were due to EC-6 and EC-7, which removed studies on fully automated pipelines without explicit interaction design. Screening yielded 42 papers. Three rounds of forward and two rounds of backward snowballing added 8 papers. Most snowballing exclusions were EC-3 (preprints without evaluation). The final dataset comprised $N = 50$ papers from 2017–2025.

We developed a codebook using a hybrid deductive–inductive approach [46], defining high-level categories from existing taxonomies [24, 27] and conducting bottom-up open coding on 25 papers (50% to allow new codes to emerge within and beyond these categories. Both authors independently coded the remaining 25 papers. Inter-rater reliability (Cohen’s kappa coefficient [84]) averaged 0.85 (indicating “strong” agreement). This produced two taxonomies for interaction techniques and UI elements.

3.1.1 Proposed taxonomies. The finalized codebook is publicly available online³. It contains two taxonomies and 50 annotated papers. To complement the textual definitions and provide intuitive visual references for each category, we present illustrations of the interaction techniques taxonomy in Figure 5 and the UIs taxonomy in Figure 6.

3.2 Design thinking workshop

This workshop proposed an interaction framework for LLM-based system to help ontology engineers progressively define ontology scope. It involved 7 PhD researchers (2 ontology engineering experts, 3 HCI experts, and 2 with expertise in both) from 4 institutions and 5 research projects in academia, industry, or both. Participants were recruited via the authors’ professional networks and referrals to ensure relevant, diverse expertise. Building on the interaction technics and UI taxonomies in Section 3.1.1, the workshop followed the Double Diamond model [13, 32] with four phases: *discover*, *define*, *develop*, and *deliver*.

3.2.1 Step 1: discover and define phases. From our review of current LLM-based systems, ontology development and mind-mapping tools, and expert HCI and OE insights, we identified four key limitations that hinder ontology engineers in ontology boundary auditing. First, recent LLM-based CQ elicitation systems from reference

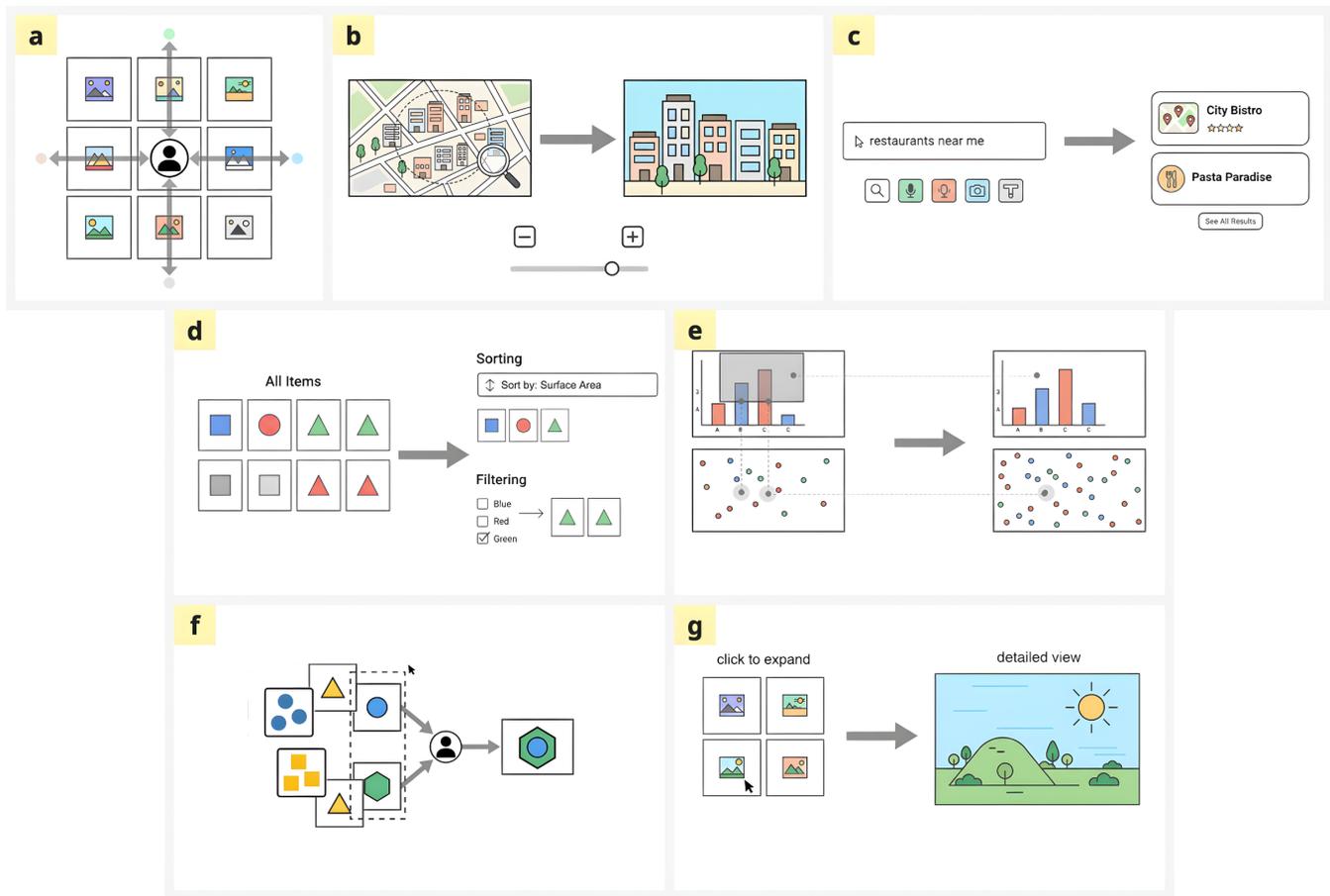


Figure 5: Illustration of interaction techniques. (a) Navigation: Users traverse the design space with directional controls while generative artificial intelligence (GenAI) generates region-specific artifacts to expand exploration and maintain spatial coherence. (b) Zooming: Users switch between global and detailed views while GenAI summarizes clusters at overview level and annotates individual elements at detail level. (c) Querying: Users issue prompts or keywords while GenAI finds and highlights semantically relevant artifacts to broaden discovery. (d) Sorting and filtering: Users reorganize or limit artifacts by criteria while GenAI suggests meaningful operations to help compare groupings and reduce complexity. (e) Linking and brushing: Users select artifacts to reveal relationships across views while GenAI infers and surfaces hidden associations and patterns. (f) Clustering and combining: Users group or merge related artifacts while GenAI proposes meaningful clusters or synthesizes composite artifacts. (g) Hover and click-to-expand: Users preview or expand artifacts while GenAI provides brief summaries for quick scanning or detailed elaborations for deeper review.

sources [6, 10, 29, 44] focus on CQ generation but offer little support for auditing large sets of CQ candidates, constraining ontology engineers' ability to expand, filter, refine CQs, and ensure they align with the intended ontology scope. Second, LLM-based CQ elicitation from domain experts [130, 138] typically uses linear, turn-taking dialogues tied to users' initial inputs [34, 42, 69, 113, 128], restricting branching into new directions and causing domain gaps to be missed, with CQs often failing to cover emerging areas needed for comprehensive ontology representation. Third, ontology development and management tools [51, 87, 89, 116, 117] that support auditing generated CQs and terminologies for scoping mainly use list-based interfaces that require users to scroll through long, nested hierarchies to find and edit terms, which hinders overview and reasoning about ontology scope [87]. Fourth, general-purpose mind-mapping tools [86, 88] offer generative support or visualizations of ontology scope but are neither designed nor empirically validated for ontology scope auditing, limiting their reliability.

3.2.2 Step 2: develop and deliver phases. These limitations led us to propose an interaction framework grounded in taxonomies of interaction techniques (Figure 5) and UIs (Figure 6) that have been validated to support both divergent and convergent thinking, to address them. This interaction framework was iteratively refined with HCI and OE experts until consensus was reached, balancing conceptual diversity and experimental feasibility. It adapts Luminat's structured generation and exploration design-space interaction framework [113] for ontology scoping by: (1) fixing the x - and y -axes to *domain coverage* and *terminology granularity*; (2) mapping generic design artifacts to CQs and their suggested terminologies; and (3) enabling auditable export of the scoped CQ space as JSON for downstream OE tasks.

3.2.3 Proposed interaction framework. Table 2 presents this proposed interaction framework.

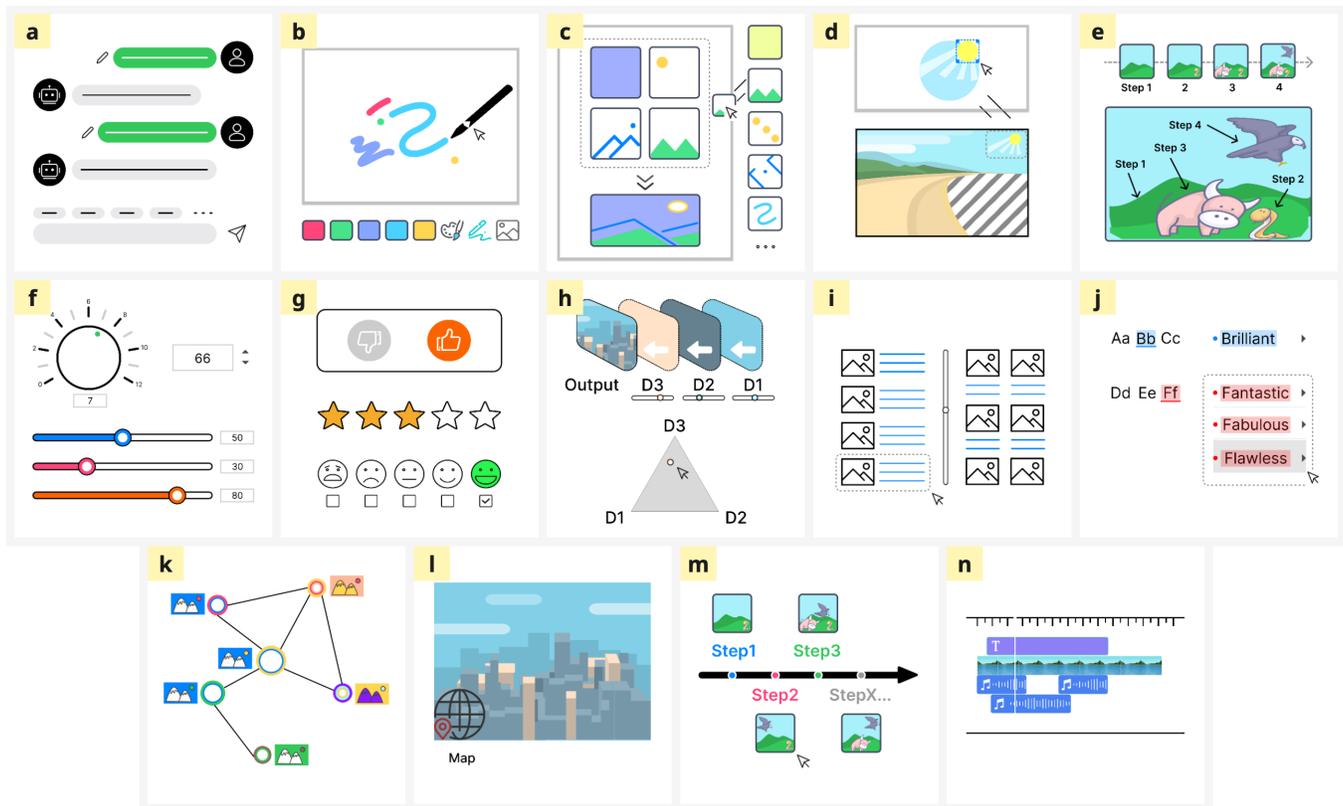


Figure 6: Illustration of UIs. (a) Chatbot: Turn-based dialogue interface where GenAI produces contextually relevant responses and suggestions for stepwise refinement. (b) Canvas: Free-form sketching workspace where GenAI produces completions or annotations for open-ended exploration. (c) Moodboard: Collage interface where GenAI produces visual variations and thematic links to foster associative thinking. (d) Focus region in context: Interface linking zoomed-in areas with global overviews, where GenAI proposes edits that balance local details and global coherence. (e) Spatially structured editor: Predefined layouts (grids, timelines) where GenAI fills or adjusts content for organized sequential development. (f) Slider, knob, and numeric textfield: Parameter controls where GenAI generates previews of changes from broad adjustments to fine-tuning. (g) Binary and Likert-scale rating: Fixed-option feedback interfaces where GenAI learns from user ratings to refine outputs and converge on preferences. (h) Interpolating slider, region, and graph: Interfaces that blend variables across dimensions while GenAI generates compositions revealing alternative possibilities. (i) List and grid: Sequential or tabular layouts where GenAI populates entries to support broad scanning and comparison. (j) In-flow options: Inline contextual alternatives where GenAI offers immediate suggestions that promote divergence without breaking workflow. (k) Node-based branches: Graphs mapping artifacts into branches while GenAI proposes new nodes or links to aid exploration and reasoning. (l) Map and earth: Spatial overviews where GenAI clusters and highlights geographic regions to direct attention to distant or overlooked options. (m) Timeline, storyline, and step sequencer: Linear layouts of ordered events where GenAI inserts or refines steps to maintain coherence. (n) Lanes and tracks: Parallel timelines where GenAI generates artifacts that integrate multiple streams for coherent refinement.

3.3 Design goals

Based on this framework, OntoScope’s design goals (DGs) are:

- **DG1 (Spatial overview):** Visualize the CQ space, showing how CQs are distributed across subdomains and levels of granularity.
- **DG2 (Divergent thinking):** Support exploratory interactions (e.g., adding new CQs, terms, or dimension values) through LLM-generated suggestions and visual feedback.
- **DG3 (Convergent thinking):** Support removing and refining similar and irrelevant CQs, terms, or dimension values to ensure precision, with the LLM learning from edits to improve future suggestions.
- **DG4 (Termination):** Export the finalized CQ space in an auditable format for reuse and integration into downstream OE tasks.

4 OntoScope

In this section, guided by the design goals, we present the OntoScope UIs and features in Section 4.1, illustrate a typical user

scenario in Section 4.2, and describe the technical implementation and prompt engineering in Section 4.3.

4.1 Interface and features

OntoScope’s interface has two parts. The first is the canvas (Fig. 7), where generated CQ candidates are clustered by terminology granularity and domain coverage. Each CQ appears as suggested terminology in the detailed view or as a node in the zoomed-out view, and users can drag and zoom to explore clusters. The second part is a set of pop-up windows (Fig. 8) that appear when users interact with canvas elements and let them add, delete, or refine dimension values, CQs, and terms.

Table 2: The interaction framework comprises UIs and interaction techniques (ITs) that support ontology engineers in ontology scoping. Each entry specifies the interaction stage, the UI or IT role, and its rationale.

Interaction stage	UI / IT	Role	Reason for inclusion
Spatial overview	Canvas UI (Spatial editor / Map)	Spatial overview and grid for placing CQs	Lets ontology engineers spot gaps or imbalances in ontology scope.
	Nodes UI Clustering and combining IT	Candidate CQs as manipulable nodes Grouping related CQs into clusters	Makes each CQ easy to inspect and compare during scoping. Reveals similarity and concentration of CQs, supporting a more balanced domain representation.
	Click-to-expand IT	Expanding a CQ node for details	Shows CQ-suggested terms for verification and refinement.
Divergent thinking	Navigation IT In-flow options UI	Moving across clusters, dimensions, and nodes Inline suggestions for new CQs, terms, or dimension values	Supports systematic exploration of the CQ space. Adds options in context so ontology engineers can expand scope without interrupting workflow.
	List UI	Structured presentation of suggestions	Clearly organizes alternatives to support comparison and informed selection.
	Zooming IT	Switching between global and detailed views	Supports moving between broad scope assessment and close inspection of specific CQ details.
Convergent thinking	Focus region in context UI Binary rating UI	Linking zoomed-in edits to global overview Deleting irrelevant CQs, terms, or dimension values	Keeps local refinements aligned with the overall ontology scope. Lets ontology engineers tighten ontology boundaries and stop similar future suggestions from system.

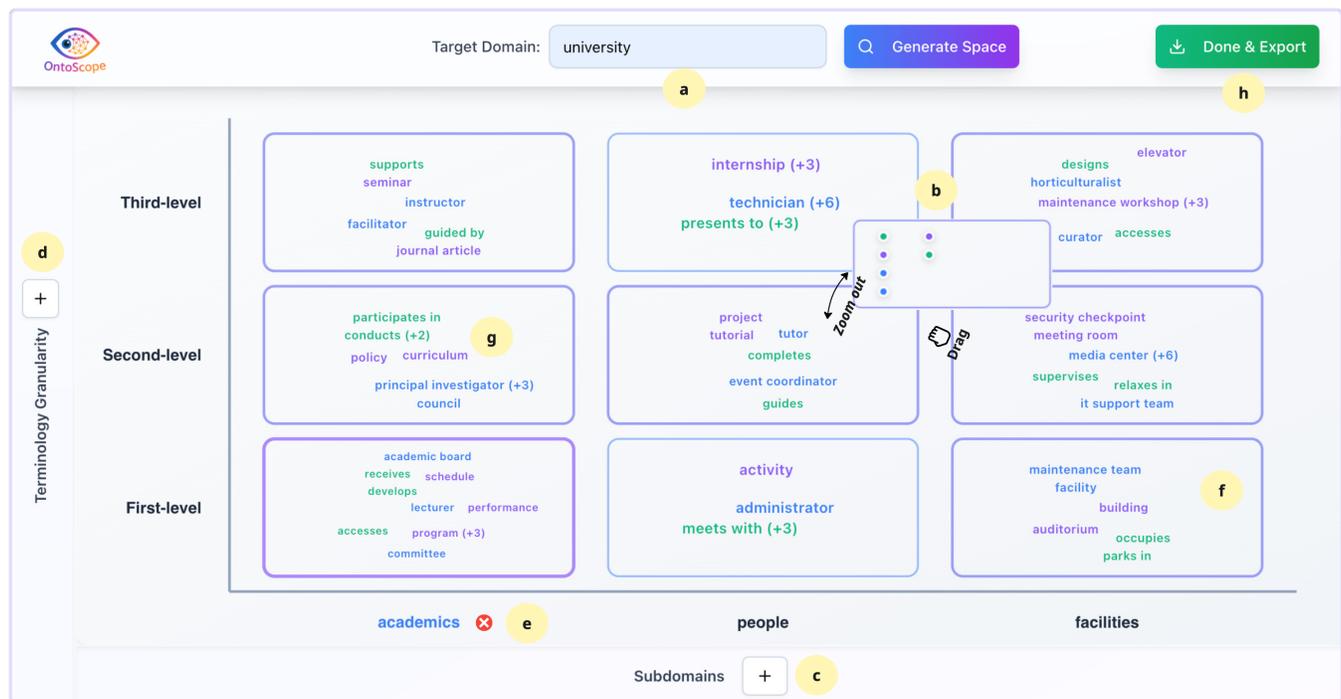


Figure 7: Main interface of OntoScope: a (DG1): domain input field to specify a target domain and generate the CQ space; b (DG1): CQ space displaying CQs as proposed terminology (one shown, others summarized as “(+n)”) clustered by terminology granularity and domain coverage. Blue marks subject CQs, green marks property CQs, and purple marks object CQs. The space supports drag and zoom and shows CQs as nodes when zoomed out; c (DG2): subdomain add button opening a pop-up to select LLM-suggested or custom subdomains; d (DG2): terminology granularity add button; e (DG3): dimension value delete button; f (DG2): clickable intersection area opening a pop-up to select LLM-suggested or custom CQs; g (DG2, DG3): clickable terminology opening a pop-up with CQ details, allowing users to edit the CQ; add LLM-suggested or custom terms; or delete the CQ and its terms; h (DG4): export button to download the refined CQ space as a JSON file for downstream tasks.

4.2 User scenario

To demonstrate how OntoScope supports ontology scoping, we describe a typical workflow. Maria, an ontology engineer developing a “university” ontology, uses OntoScope to explore, refine, and finalize CQs, terms, and dimension values. She enters *university* in the domain input box (Fig. 7a), and OntoScope generates a CQ

space (Fig. 7b) with three subdomains (*academics*, *people*, *facilities*) crossed with three terminology granularity levels. Zooming out, she notices missing subdomains like *research* and *services*. She clicks the subdomain add button (Fig. 7c), which opens the dimension value popup. There, she enters custom values (Fig. 8b) and selects

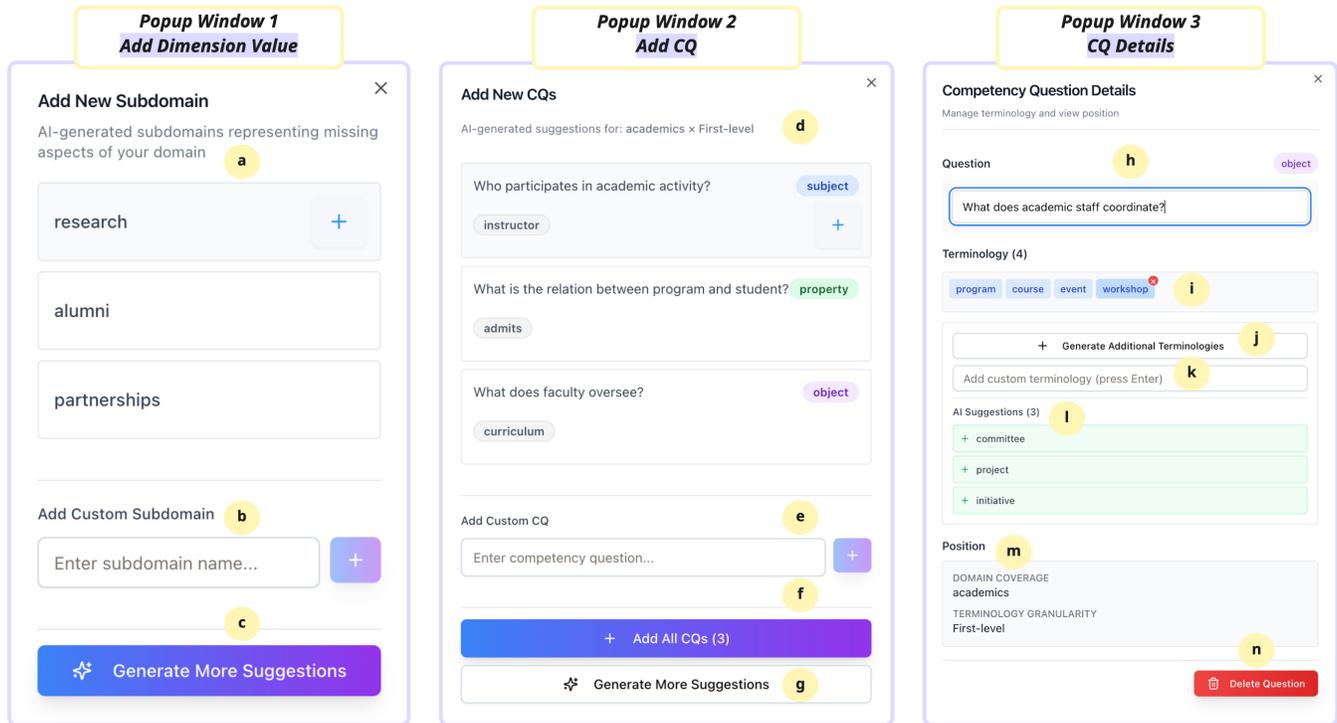


Figure 8: Popup windows in OntoScope: Popup Window 1 (DG2) lets users add dimension values via LLM suggestions (a), a custom-value text box (b), and a more-suggestions button (c). Popup Window 2 (DG2) lets users add CQs via LLM suggestions (d), a custom-CQ text box (e), an add-all-suggestions button (f), and a more-suggestions button (g). Popup Window 3 (DG2, DG3) shows CQ details, including a CQ edit box (h); buttons to delete terms (i) and generate more terms (j); a custom-terms text box (k); an LLM-suggested terms list (l); a panel showing the CQ's position in the space (m); and a CQ delete button (n).

relevant LLM suggestions (Fig. 8a, c) until they become loosely related. Maria then populates CQs systematically across the grid. She clicks the *academics* × *first-level* intersection (Fig. 7f) to open the CQ popup. She enters custom terms (Fig. 8e) and selects relevant suggestions (Fig. 8d, f, g) until they match existing CQs.

After populating all existing intersections, she adds a fourth level to test if finer granularity is needed (Fig. 7d), but finds the CQ suggestions too detailed and discards it. She then edits CQs-suggested terms by clicking each one of them (Fig. 7g) to open the CQ detail popup. There, she enters custom terms (Fig. 8k), generates term suggestions (Fig. 8j), and accepts non-overlapping ones (Fig. 8l) until they become variants. She then zooms out to review the full space, deletes unnecessary subdomains and granularity levels (Fig. 7e), removes similar CQs (Fig. 8n) and terms (Fig. 8i), and refines unsatisfied CQs (Fig. 8h) until satisfied with the space. Finally, she exports this space as a JSON file (Fig. 7h), representing the ontology scope for downstream OE tasks.

4.3 Implementation and prompt engineering

OntoScope's frontend uses React with Vite for fast, component-based development. Styling uses Tailwind CSS and Shadcn/ui components built on Radix UI primitives for accessible, responsive design. Interactive visualization uses D3.js for dynamic scatter plots. The backend uses Node.js and Express.js to provide a RESTful API for real-time LLM-assisted interactions. The system uses OpenAI's

gpt-4.1⁶ to generate candidate CQs, dimension values, and terminology. In pilot tests, gpt-4.1 outperformed gemini-2.5-pro⁷ and claude-opus-4-20250514⁸ on CQs well-formedness, terminological precision, and content relevance. For CQs generation, the model is instructed to treat singular and plural forms carefully, as they may yield different answers. For CQ, term, and dimension suggestions, OntoScope uses an automatic regeneration pipeline to remove redundancy within suggestions and against existing content. The pipeline regenerates candidates until it reaches a duplicate-free set, for up to ten iterations, since additional iterations rarely add valid suggestions but significantly increase latency.

5 User evaluation

This evaluation examines how well the proposed interaction framework helps ontology engineers define an ontology's scope, focusing on three evaluation questions (EQs):

- **EQ1:** How useful is the interaction framework in supporting divergent and convergent thinking during domain boundary auditing?
- **EQ2:** How useful is the set of CQs resulting from the auditing process for defining the scope of the ontology?
- **EQ3:** Does support for divergent and convergent thinking during boundary auditing positively contribute to defining the scope of the ontology?

⁶<https://openai.com/index/gpt-4-1/>

⁷<https://modelcards.withgoogle.com/assets/documents/gemini-2.5-pro.pdf>

⁸<https://docs.claude.com/en/docs/about-claude/models/overview>

5.1 Participants

We recruited 15 ontology engineers (9 female, 6 male; P1–P15; age: $M = 26.87$, $SD = 7.57$) who had not participated in the design thinking workshop. The sample included 9 PhD researchers, 1 with a master’s degree, and 5 undergraduates; 8 worked in academia, 4 in industry, and 3 in both. All volunteered and gave informed consent. The study was approved by the first author’s university ethics committee. Figure 9 shows participants’ self-reported familiarity with knowledge elicitation, CQs, and LLMs in OE, as well as their OE application domains. These demographics reflect diverse expertise, suggesting our sample is representative.

Recruitment was conducted via the W3C Semantic Web public mailing list⁹. The full recruitment details are in Appendix B.1. Our sample size ($N=15$) is comparable to prior evaluations of ontology tools [130, 138]. Some participants had limited familiarity with knowledge elicitation or CQs, reflecting the range of specializations in OE. At the start of each session, we introduced CQs and their role in OE.

5.2 Procedure

All 15 sessions were conducted online via Microsoft Teams. After obtaining consent, participants completed a demographic questionnaire, received a 10-minute OntoScope tutorial following a scripted protocol, and performed a 20-minute concurrent think-aloud [119, 136] using OntoScope to define boundaries for one familiar domain from the WDV-CQ-HA dataset [29] (this dataset provides CQs with expert-validated answers across 20 broadly accessible Wikidata domains and served as the benchmark for user-generated CQs and answers). A pilot with 3 ontology engineers refined the protocol, OntoScope parameters, and usability features (the full pilot study details are in Appendix B.2). Sessions concluded with a post-task questionnaire and retrospective think-aloud. The full procedure details are in Appendix B.3.

5.2.1 Data analysis. All audio recordings were auto-transcribed and manually corrected. Identifying details were removed before secure storage. For **EQ1**, we conducted thematic analysis using open and axial coding, triangulated with Creativity Support Index (CSI) scores [28] from the post-task questionnaire. CSI is a validated scale for perceived creativity support, unlike general usability or workload measures such as the System Usability Scale (SUS) [26] or the NASA Task Load Index (NASA-TLX) [60], which focus on efficiency, usability, or workload rather than creative processes.

For **EQ2**, we computed the F_1 score of participant-generated terminologies against the WDV-CQ-HA dataset [29] (this dataset provides CQs with expert-validated answers across 20 broadly accessible Wikidata domains). F_1 balances precision and recall: high precision keeps the scope from becoming too broad, while high recall keeps it from becoming too narrow. To calculate precision and recall, we first counted correctly matched terms (true positives, TP), using `gemini-embedding-001` [79] (the full model selection details are in Appendix B.4) to find the best-matching gold standard term for each user-generated term in its domain. Each match was manually validated. We then defined false positives (FP) as user-generated terms not matched to any gold standard term and false negatives

(FN) as gold standard terms not matched by any user-generated term. Precision and recall were calculated as $\text{Precision} = \frac{TP}{TP+FP}$ and $\text{Recall} = \frac{TP}{TP+FN}$, and the F1 score as $F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$.

For **EQ3**, we examined how participants’ CSI scores relate to their F_1 scores. If OntoScope supports both divergent thinking (broad exploration yielding high recall) and convergent thinking (targeted refinement yielding high precision), then participants with higher CSI scores should also achieve higher F_1 scores, producing a positive correlation.

5.3 Results

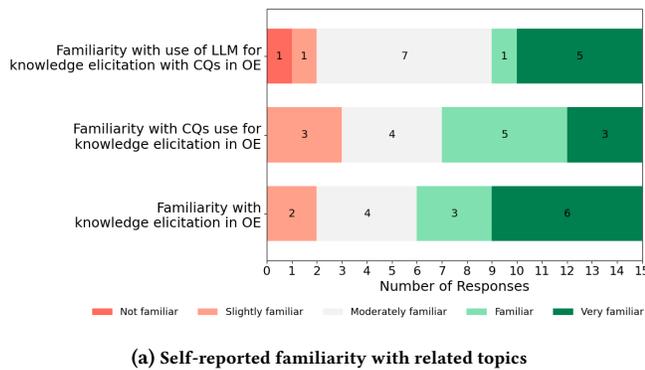
5.3.1 How useful is the interaction framework in supporting divergent and convergent thinking during domain boundary auditing? (EQ1). In Table 3, participants achieved an average CSI score of 81.57 ($SD = 8.77$) out of 100, indicating that OntoScope was perceived as highly useful for supporting divergent and convergent thinking during ontology scoping. The CSI includes five factors, each measured by two items on a 1–10 scale, with higher scores indicating stronger perceived support (Fig. 10). The highest average scores were for *Exploration* (17.27/20, $SD = 2.34$) and *Expressiveness* (17.07/20, $SD = 2.49$). The high exploration score suggests that OntoScope lowered barriers to divergent thinking by continually generating new CQs, terms, and axis values, prompting participants to consider alternative domain dimensions. As one participant (PID 07) noted, “I feel like this system can just keep generating CQs, and it made me curious about what other directions I hadn’t even thought of yet.” The high expressiveness score indicates that participants retained fine-grained control over the suggestions, which is critical for convergence when terminology must precisely reflect domain knowledge. As another participant (PID 12) remarked, “This system really gave me confidence in the outcome, because all the decisions felt like they were mine. If I didn’t like something, I could just click and change it right away.”

Table 3: CSI results based on $n = 15$ participants (computed with 5 factors, normalized by 2).

Factor	Avg. Score (SD)	Avg. Factor Count
Exploration	17.27 (2.34)	3.20
Enjoyment	15.93 (2.69)	1.33
Expressiveness	17.07 (2.49)	2.40
Results Worth Effort	15.80 (3.00)	2.20
Immersion	12.93 (4.04)	0.87
Overall CSI Score	81.57 (8.77)	

Two of the three remaining factors also scored highly: *Enjoyment* (15.93/20, $SD = 2.69$) and *Results Worth Effort* (15.80/20, $SD = 3.00$). Participants found OntoScope engaging and productive. Enjoyment came from immediate, varied contents (CQs, terms, and dimension values) suggestions that sustained curiosity by continually introducing new directions. As one participant (PID 05) noted, “I enjoyed clicking through and seeing new CQs come up; it kept me curious about what would appear next.” Perceived worthwhileness stemmed from seeing clear progress in defining the ontology scope. As another participant (PID 14) noted, “I felt my time was well

⁹<https://lists.w3.org/Archives/Public/semantic-web/>



(a) Self-reported familiarity with related topics



(b) Domains covered by participants' OE projects

Figure 9: Participant demographics distribution

spent because I could see the ontology scope taking shape step by step.” Immersion received an average score of 12.93/20 ($SD = 4.04$), indicating that participants could focus on ontology scoping while remaining aware of the system when needed. For example, one participant (PID 08) remarked, “OK, too many CQs around here, so, um, let me have a look at them first.” and then dragged the canvas to reorganize the view and track scattered CQs. We interpret this not as reduced immersion, but as active alternation between exploring new CQs and consolidating existing ones, a process essential for effective domain boundary auditing.

We evaluated preferences among the five CSI factors using paired comparisons to identify which participants saw as most important for effective ontology scoping. The results (Table 3, Fig. 11) show that *Exploration* had the highest average factor count (3.20), indicating that participants viewed divergent exploration as central to effective scoping, since generating many possibilities supports later refinement. As PID 04 said, “It feels like I can just keep opening up new directions, and I don’t want to miss any of them.” *Expressiveness* (2.40) and *Results Worth Effort* (2.20) followed, suggesting that after generating many CQs, participants prioritized refining them and seeing concrete progress. As PID 11 noted, “I kept tweaking the terms a bit here and there, and after a while I could see the scope really shaping up.” *Enjoyment* (1.33) and *Immersion* (0.87) were less preferred, indicating that ontology engineers emphasized instrumental over hedonic qualities, valuing valid results from strong support for exploration and refinement more than enjoyment or absorption. As PID 06 commented, “Yeah, I am enjoy to use it... like the current interaction is good... but what I really want to see is that my scope getting more correct.”

To assess CSI robustness, we examined correlations between the two statements in each factor (Fig. 12). All five factors showed strong positive correlations ($r = 0.70$ – 0.78 , all $p < 0.01$), indicating that each pair measured the same construct and that CSI scores reliably reflect OntoScope’s support for divergent and convergent thinking in ontology scoping.

5.3.2 How useful is the set of CQs resulting from the auditing process for defining the scope of the ontology? (EQ2). The F_1 score distribution (Fig. 13a) has a mean of 0.66 and a median of 0.77, indicating that auditing CQs supported precise terminology and broad domain coverage. Participants’ reflections echo this. PID 07

noted that “the terms it suggested always felt more exact than what I would have come up with myself, almost like it expressed what I wanted to say but couldn’t quite put into words.” The suggestions shifted participants from vague descriptions to precise, domain-appropriate labels, improving alignment with the gold standard. PID 12 explained that “without it I would have struggled to define the domain properly, since each recommendation reminded me of something I had overlooked before.” The recommendations acted as recall prompts, helping participants include overlooked domain aspects, cover more reference terms, and strengthen the precision–coverage balance reflected in their F_1 scores. The four lower scores came from participants who focused on niche domain aspects not fully represented in the gold standard but later validated by ontology and domain experts. These were not weaker performances but complementary perspectives that extended the domain beyond the predefined reference set. PID 10 reflected: “Whenever a new subdomain came up it still felt connected, even if it wasn’t quite what I had in mind. It made me see how this domain can be looked at from so many angles, and it kept nudging me to notice things I might’ve skipped. By the end, I felt confident I’d mapped out the scope well.”

5.3.3 Does support for divergent and convergent thinking during boundary auditing positively contribute to defining the scope of the ontology? (EQ3). The correlation between CSI and F_1 (Fig. 13c) is strongly positive ($r = 0.76$, $p < 0.01$): participants who engaged more in divergent and convergent thinking used terminology more precisely and covered more of the ontology. This suggests that OntoScope’s dimensions, intersections, CQs, and terminologies effectively supported exploration and refinement. As PID 14 noted, “Without seeing how different dimensions could intersect, I don’t think I would’ve thought that far. It pointed me to new areas I hadn’t considered, and once I started adding CQs and terms there, I could clearly see what was already covered and what was still missing. That made me feel like my scope was really moving in a better direction.” The interface thus externalised possibilities, reduced working-memory load, and let participants keep more candidate terms in play while exploring the domain from multiple perspectives.

OntoScope supports divergence by widening the search space. Each dimension and CQ serves as a cognitive cue, prompting recall

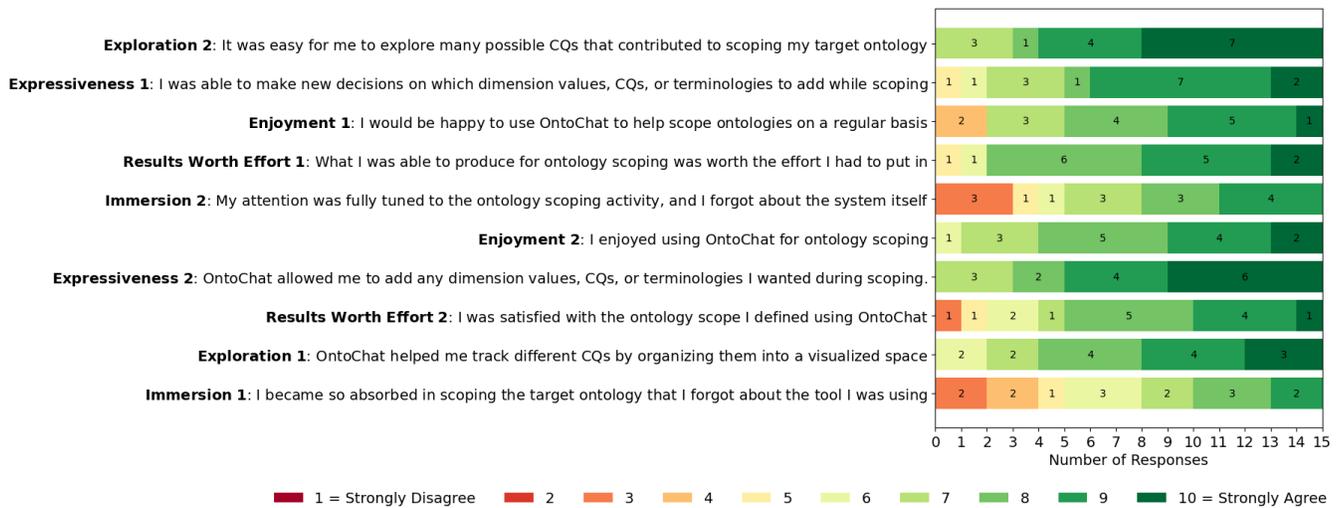


Figure 10: Bar chart of participant ratings for ten CSI statements across five factors (Enjoyment, Exploration, Expressiveness, Immersion, Results Worth Effort) on a 10-point Likert scale.

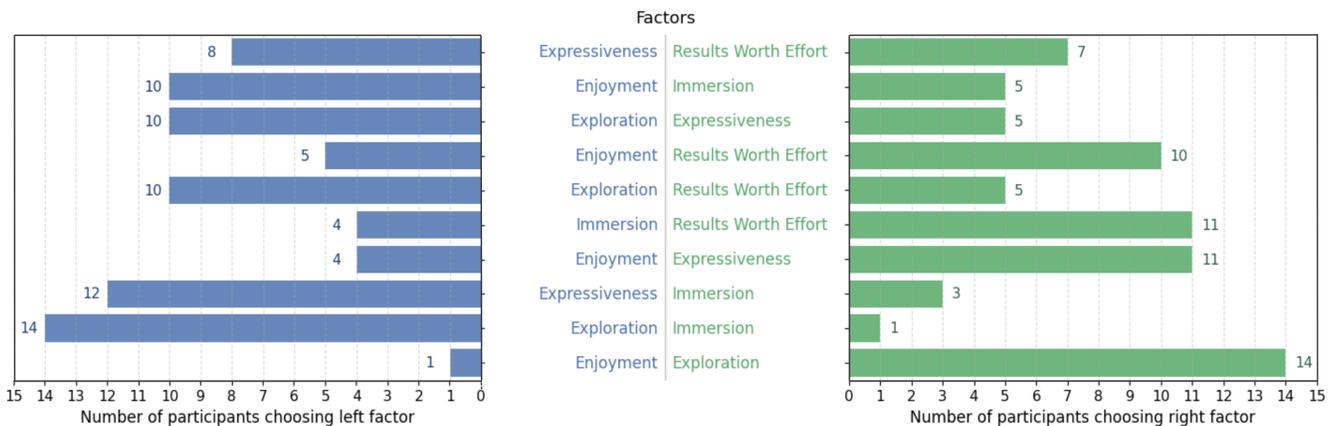


Figure 11: Results of the CSI paired comparison task. Each row compares two factors: blue bars show selections for the left factor, green bars for the right factor, and factor names appear in the centre column.

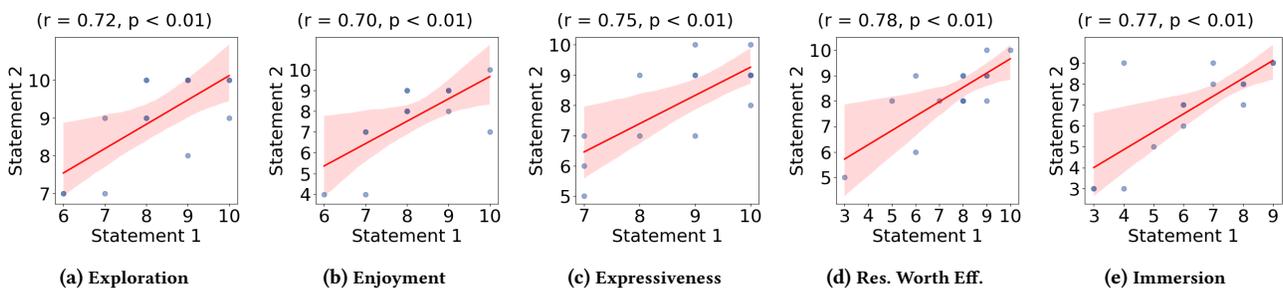


Figure 12: Reliability analysis of CSI factors. Subfigures (a–e) show scatter plots for each factor with blue data points, a red regression line, and a shaded red 95% confidence interval. Pearson’s r and p values indicate correlation strength and significance. “Res. Worth Eff.” denotes *Results Worth Effort*.

of related but non-obvious concepts and broadening participants’ domain perspective. PID 05 noted, “Each time a new suggestion popped up, it reminded me of a part of the domain I hadn’t thought

about yet, and suddenly I realised there were many more angles I could take.” Intersections anchored these cues in specific contexts,

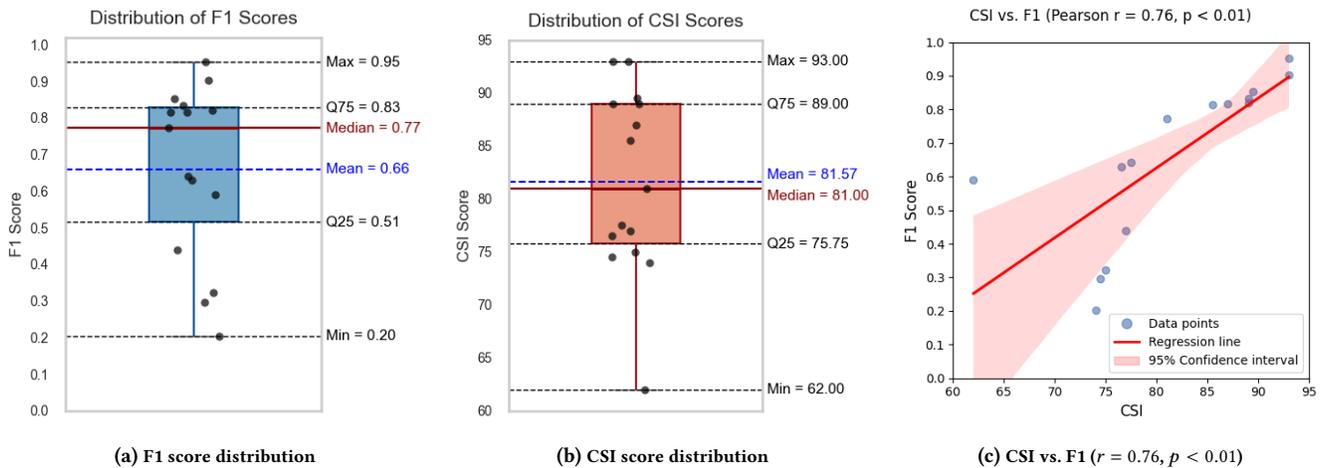


Figure 13: Comparison of F1 and CSI scores. (a) Boxplot of F1 scores. (b) Boxplot of CSI scores. (c) Scatter plot of CSI vs. F1 with regression line and 95% confidence interval, indicating a strong positive correlation.

so coverage expanded in a controlled, not overwhelming, way. OntoScope channelled divergent thinking into structured exploration, helping participants uncover more diverse subdomains. Once a richer set of candidate terms was established, OntoScope supported convergence. Intersections revealed gaps and overlaps, helping participants identify omissions, refine or remove similar CQs, terms, or dimension values, and align scope. PID 09 said, “Seeing terms side by side made it obvious when two were too close, so I could pick the sharper one. It also showed me which parts were still empty, so I knew exactly where to focus next.” Term suggestions nudged participants from vague words to precise, domain-appropriate labels. This iterative cycle of suggestion, checking, and revision reduced noise, leaving only in-scope elements.

6 Discussion

We evaluate our design goals in Section 6.1. Our OE-specific contributions are in Section 6.2, and our broader impact on knowledge-intensive works appears in Section 6.3.

6.1 Reflection on design goal fulfilment

DG1 (Spatial overview) externalized the CQ space on a manipulable canvas, lowering memory load and revealing coverage gaps. High Exploration scores (17.27/20, factor count = 3.20) show this layout supported systematic exploration. DG2 (Divergent thinking) enabled exploration through on-demand contextual LLM suggestions, avoiding upfront overload. The strong CSI–F1 correlation ($r=0.76, p<0.01$) indicates this divergent support improved domain coverage: more systematic exploration yielded more gold standard terms. DG3 (Convergent thinking) balances divergence by immediately editing or removing CQs, terms, and dimensions, with the LLM learning from corrections to improve future suggestions. The high Expressiveness score (17.07/20) and median F1 of 0.77 suggest this bidirectional loop effectively supported terminological precision. DG4 (Termination) enabled closure via export, and Results Worth Effort scores (15.80/20) show participants saw their scoping work as producing reusable artifacts.

6.2 From fragmented workarounds to a unified process for ontology scoping

Ontology scoping has long been fragmented across multiple tools and ad hoc workarounds, with no widely accepted framework or seamless toolchain [111, 121]. Requirements are often gathered in generic tools like text editors and spreadsheets, while management and auditing occur separately in tools such as Protégé [87]. This split creates technical and resource burdens [130], causing inefficiencies and inconsistencies, especially at scale. Our interaction framework embeds ontology scoping in a unified workflow where users explore and refine candidate requirements in a visual workspace, producing reliable scope representations directly usable for ontology modelling. In OE, it works with the aim of providing a unified path from requirements elicitation to ontology construction.

6.3 Towards designing transparent human–AI interaction for knowledge-intensive work

Knowledge-intensive work requires iterative information exploration and synthesis, yet most GenAI tools support only single prompt–response exchange, limiting users’ ability to externalize intermediate reasoning, track assumptions, and build shared understanding [49, 65, 70, 129]. This is especially problematic when people must audit or defend AI-generated content, because opaque generation undermines accountability and hinders verification of provenance or contestation of flawed reasoning [66, 81, 93]. Natural language interfaces broaden access but also introduce prompt brittleness, narrow context windows, and poor traceability, obscuring how inputs shape outputs [65]. The divergent–convergent interaction framework addresses these issues by making AI reasoning explicit within structured design spaces and treating model outputs as provisional artifacts for exploration, comparison, and refinement. Divergent thinking surfaces multiple alternatives (e.g., rival hypotheses or design variants), while convergent thinking supports evaluation and synthesis into integrated outcomes [42]. Organizing

alternatives along salient dimensions and inspecting their intersections helps users navigate heterogeneous information spaces [65, 113, 114] while auditing the reasoning that produced them. Thus, this interaction framework recasts GenAI tools as transparent cognitive collaborators that scaffold reasoning while preserving accountability, advancing human-centered AI that couples generative capacity with interpretability, contestability, and traceability [66, 81, 93].

7 Conclusions

This paper presents an interaction framework that helps ontology engineers audit ontology scope using large sets of LLM-generated candidate CQs. We derived this interaction framework by combining a literature search with a design thinking workshop, integrating interaction technics and UIs that support divergent and convergent thinking for domain boundary auditing. We then built OntoScope, an LLM-based interactive system that visualizes candidate CQs by domain coverage and terminology granularity, enabling ontology engineers to iteratively expand, refine, or narrow scope. In a user study with 15 ontology engineers, this interaction framework supported both divergent and convergent thinking; deeper engagement yielded more precise terminology and broader ontology scope coverage. Limitations include the small sample and lack of comparison with baseline AI tools; future work should use larger samples (including domain experts working in OE) and controlled comparisons to evaluate OntoScope and demonstrate its advantages. To our knowledge, this is the first validated interaction framework with an LLM-based system for ontology domain boundary auditing, and the first to unify fragmented, expert-dependent scoping practices into an accessible approach. Its design also shows how LLM-based systems can transparently and accountably support broader knowledge-intensive tasks for the HCI and OE communities.

Acknowledgments

This study was supported by several sources, including the Engineering and Physical Sciences Research Council [grant number EP/Y009800/1], funded through Responsible AI UK (KP0011); SIEMENS AG and Institute for Advanced Study, the Technical University of Munich, Germany. We'd also like to thank all the evaluators who provided helpful feedback on the system.

References

- [1] Abdelghany Salah Abdelghany, Nagy Ramadan Darwish, and Hesham Ahmed Hefni. 2019. An agile methodology for ontology development. *International Journal of Intelligent Engineering and Systems* 12, 2 (2019), 170–181.
- [2] Kat Agres, Stephen McGregor, Matthew Purver, and Geraint A Wiggins. 2015. Conceptualizing Creativity: From Distributional Semantics to Conceptual Spaces. In *ICCC*. Association for Computational Creativity, Park City, Utah, USA, 118–125.
- [3] Christopher Ahlberg, Christopher Williamson, and Ben Shneiderman. 1992. Dynamic queries for information exploration: An implementation and evaluation. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. Association for Computing Machinery, New York, NY, USA, 619–626.
- [4] Reham Alharbi, Valentina Tamma, and Floriana Grasso. 2021. Characterising the gap between theory and practice of ontology reuse. In *Proceedings of the 11th Knowledge Capture Conference*. Association for Computing Machinery, New York, NY, USA, 217–224.
- [5] Reham Alharbi, Valentina Tamma, and Floriana Grasso. 2024. Requirement-based methodological steps to identify ontologies for reuse. In *International Conference on Advanced Information Systems Engineering*. Springer, Springer Nature Switzerland, Cham, 64–72.
- [6] Reham Alharbi, Valentina Tamma, Floriana Grasso, and Terry Payne. 2024. An experiment in retrofitting competency questions for existing ontologies. In *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*. Association for Computing Machinery, New York, NY, USA, 1650–1658.
- [7] Reham Alharbi, Valentina Tamma, Floriana Grasso, and Terry R Payne. 2024. Investigating open source LLMs to retrofit competency questions in ontology engineering. In *Proceedings of the AAAI Symposium Series*. AAAI Press, Washington, DC, USA, 188–198.
- [8] Reham Alharbi, Valentina Tamma, Floriana Grasso, and Terry R Payne. 2024. A review and comparison of competency question engineering approaches. In *International Conference on Knowledge Engineering and Knowledge Management*. Springer, Springer Nature Switzerland, Cham, 271–290.
- [9] Mary-Jane Antia and C Maria Keet. 2021. 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)*. Special Interest Group on Controlled Natural Language, Amsterdam, Netherlands, 9 pages.
- [10] Mary-Jane Antia and C Maria Keet. 2023. Automating the generation of competency questions for ontologies with AgOCQs. In *Iberoamerican Knowledge Graphs and Semantic Web Conference*. Springer, Springer Nature Switzerland, Cham, 213–227.
- [11] Sabrina Azzi, Ali Assi, and Stéphane Gagnon. 2022. Scoring ontologies for reuse: an approach for fitting semantic requirements. In *Research Conference on Metadata and Semantics Research*. Springer, Springer Nature Switzerland, Cham, 203–208.
- [12] John Baer. 2014. *Creativity and divergent thinking: A task-specific approach*. Psychology Press, United States of America.
- [13] Bela H Banathy. 2013. *Designing social systems in a changing world*. Springer Science & Business Media, New York.
- [14] Lyn Bartram, Albert Ho, John Dill, and Frank Henigman. 1995. 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*. Association for Computing Machinery, New York, NY, USA, 207–215.
- [15] Michel Beaudouin-Lafon and Wendy E Mackay. 2007. Prototyping tools and techniques. In *The human-computer interaction handbook*. CRC Press, United States, 1043–1066.
- [16] Benjamin B Bederson and James D Hollan. 1994. Pad++ a zooming graphical interface for exploring alternate interface physics. In *Proceedings of the 7th annual ACM symposium on User interface software and technology*. Association for Computing Machinery, New York, NY, USA, 17–26.
- [17] SHNEIDERMAN Ben. 1996. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings of the IEEE Symposium on Visual Languages (VL'96)*. IEEE Computer Society, USA, 8 pages.
- [18] Camila Bezerra and Fred Freitas. 2017. Verifying Description Logic Ontologies based on Competency Questions and Unit Testing. *ONTOBRAS 17 (2017)*, 159–164.
- [19] Camila Bezerra, Filipe Santana, and Fred Freitas. 2014. CQChecker: a tool to check ontologies in OWL-DL using competency questions written in controlled natural language. *Learning & Nonlinear Models* 12, 2 (2014), 4.
- [20] Michael Mose Biskjaer, Peter Dalsgaard, and Kim Halskov. 2014. A constraint-based understanding of design spaces. In *Proceedings of the 2014 conference on Designing interactive systems*. Association for Computing Machinery, New York, NY, USA, 453–462.
- [21] Margaret A Boden. 2010. *Creativity and art: Three roads to surprise*. Oxford University Press, USA.
- [22] Jan O Borchers. 2000. A pattern approach to interaction design. In *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques*. Association for Computing Machinery, New York, NY, USA, 369–378.
- [23] Stephen Brake, Bryan Wang, Mauricio Sousa, Sageev Oore, and Tovi Grossman. 2023. 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*. Association for Computing Machinery, New York, NY, USA, 1–14.
- [24] Amani Braham, Félix Buendía, Maha Khemaja, and Faiez Gargouri. 2022. User interface design patterns and ontology models for adaptive mobile applications. *Personal and Ubiquitous Computing* 26, 6 (2022), 1395–1411.
- [25] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.
- [26] John Brooke et al. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.
- [27] Frederik Brudy, Christian Holz, Roman Rädle, Chi-Jui Wu, Steven Houben, Clemens Nylandsted Klokmose, and Nicolai Marquardt. 2019. 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. Association for Computing Machinery, New York, NY, USA, 1–28.
- [28] Erin Cherry and Celine Latulipe. 2014. Quantifying the creativity support of digital tools through the creativity support index. *ACM Transactions on Computer-Human Interaction (TOCHI)* 21, 4 (2014), 1–25.
- [29] Fiorela Ciroku, Jacopo de Berardinis, Jongmo Kim, Albert Meroño-Peñuela, Valentina Presutti, and Elena Simperl. 2024. RevOnt: Reverse engineering of competency questions from knowledge graphs via language models. *Journal of Web Semantics* 82 (2024), 100822.
- [30] Nancy J Cooke. 1999. Knowledge elicitation. In *Handbook of Applied Cognition*. John Wiley & Sons Ltd, UK, 479–509.
- [31] Robert A Cortes, Adam B Weinberger, Richard J Daker, and Adam E Green. 2019. Re-examining prominent measures of divergent and convergent creativity. *Current Opinion in Behavioral Sciences* 27 (2019), 90–93.
- [32] Design Council. 2005. The Double Diamond: A universally accepted depiction of the design process. <https://www.designcouncil.org.uk/our-resources/the-double-diamond/> Accessed: April 28, 2025.
- [33] Arthur Cropley. 2006. In praise of convergent thinking. *Creativity research journal* 18, 3 (2006), 391–404.
- [34] Nigel Cross. 2004. Expertise in design: an overview. *Design studies* 25, 5 (2004), 427–441.
- [35] Jacopo de Berardinis, Valentina Anita Carriero, Nitisha Jain, Nicolas Lazzari, Albert Meroño-Peñuela, Andrea Poltronieri, and Valentina Presutti. 2023. The polifonia ontology network: Building a semantic backbone for musical heritage. In *International Semantic Web Conference*. Springer, Springer Nature Switzerland, Cham, 302–322.
- [36] Edward De Bono and Efrim Zimbalist. 1970. *Lateral thinking*. Penguin London, UK.
- [37] Antonio De Nicola and Michele Missikoff. 2016. A lightweight methodology for rapid ontology engineering. *Commun. ACM* 59, 3 (2016), 79–86.
- [38] Antonio De Nicola, Michele Missikoff, and Roberto Navigli. 2005. A proposal for a unified process for ontology building: UPON. In *International conference on database and expert systems applications*. Springer, Springer-Verlag, Berlin, Heidelberg, 655–664.
- [39] Hérie B de Vries and Todd I Lubart. 2019. Scientific creativity: divergent and convergent thinking and the impact of culture. *The Journal of Creative Behavior* 53, 2 (2019), 145–155.
- [40] Nicola Dell, Vidya Vaidyanathan, Indrani Medhi, Edward Cutrell, and William Thies. 2012. "Yours is better!" participant response bias in HCI. In *Proceedings of the sigchi conference on human factors in computing systems*. Association for Computing Machinery, New York, NY, USA, 1321–1330.
- [41] Jill R Demirci. 2024. About research: conducting better qualitative interviews. *Journal of Human Lactation* 40, 1 (2024), 21–24.
- [42] Steven P Dow, Alana Glasco, Jonathan Kass, Melissa Schwarz, Daniel L Schwartz, and Scott R Klemmer. 2010. Parallel prototyping leads to better design results, more divergence, and increased self-efficacy. *ACM Transactions on Computer-Human Interaction (TOCHI)* 17, 4 (2010), 1–24.
- [43] Paola Espinoza-Arias, Daniel Garijo, and Oscar Corcho. 2022. Extending ontology engineering practices to facilitate application development. In *International Conference on Knowledge Engineering and Knowledge Management*. Springer, Springer-Verlag, Berlin, Heidelberg, 19–35.
- [44] Nadeen Fathallah, Arunav Das, Stefano De Giorgis, Andrea Poltronieri, Peter Haase, and Liubov Kovriguina. 2024. Neon-GPT: a large language model-powered pipeline for ontology learning. In *European Semantic Web Conference*. Springer, Springer Nature Switzerland, Cham, 36–50.
- [45] Yingchaojie Feng, Xingbo Wang, Kam Kwai Wong, Sijia Wang, Yuhong Lu, Minfeng Zhu, Baicheng Wang, and Wei Chen. 2023. Promptmagician: Interactive prompt engineering for text-to-image creation. *IEEE Transactions on Visualization and Computer Graphics* 30, 1 (2023), 295–305.
- [46] Jennifer Fereday and Eimear Muir-Cochrane. 2006. 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] George W Furnas. 1986. Generalized fisheye views. *Acm Sigchi Bulletin* 17, 4 (1986), 16–23.
- [48] Berys Gaut. 2003. Creativity and Imagination. In *The Creation of Art: New Essays in Philosophical Aesthetics*. Berys Gaut and Paisley Livingston (Eds.). Cambridge University Press, Cambridge, 148–173.
- [49] Michail Giannakos, Roger Azevedo, Peter Brusilovsky, Mutlu Cukurova, Yannis Dimitriadis, Davinia Hernandez-Leo, Sanna Järvelä, Manolis Mavrikis, and Bart Rienties. 2025. The promise and challenges of generative AI in education. *Behaviour & Information Technology* 44, 11 (2025), 2518–2544.
- [50] Ken J Gilhooly, George Georgiou, and Ultan Devery. 2013. Incubation and creativity: Do something different. *Thinking & Reasoning* 19, 2 (2013), 137–149.
- [51] PoolParty / GraphAware GmbH. 2025. *PoolParty Overview*. PoolParty. <https://help.poolparty.biz/en/poolparty-overview.html> Accessed 2025-10-08.
- [52] Gabriela Goldschmidt. 2016. Linkographic evidence for concurrent divergent and convergent thinking in creative design. *Creativity research journal* 28, 2 (2016), 115–122.
- [53] Asunción Gómez-Pérez, Mariano Fernández-López, and Oscar Corcho. 2004. *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer, London.
- [54] Saul Greenberg, Sheelagh Carpendale, Nicolai Marquardt, and Bill Buxton. 2012. *Sketching user experiences: The workbook*. Elsevier, USA.
- [55] Michael Grüninger and Mark S. Fox. 1995. Methodology for the design and evaluation of ontologies. In *Proc. IJCAI'95, Workshop on Basic Ontological Issues in Knowledge Sharing*. Morgan Kaufmann, Canada, 10 pages.
- [56] Michael Grüninger and Mark S Fox. 1995. The role of competency questions in enterprise engineering. In *Benchmarking—Theory and practice*. Springer, Boston, MA, 22–31.
- [57] Nicola Guarino. 1998. *Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy*. Vol. 46. IOS press, Italy.
- [58] Joy Paul Guilford. 1961. Three faces of intellect. *American Psychologist* 14, 8 (1961), 469–479.
- [59] Kim Halskov and Caroline Lundqvist. 2021. Filtering and informing the design space: Towards design-space thinking. *ACM Transactions on Computer-Human Interaction (TOCHI)* 28, 1 (2021), 1–28.
- [60] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Human Mental Workload*, Peter A. Hancock and Najmedin Meshkati (Eds.). Advances in Psychology, Vol. 52. North-Holland, Amsterdam, Netherlands, 139–183. doi:10.1016/S0166-4115(08)62386-9
- [61] Aron Henriksson, Hans Moen, Maria Skeppstedt, Vidas Daudaravičius, and Martin Duneld. 2014. Synonym extraction and abbreviation expansion with ensembles of semantic spaces. *Journal of biomedical semantics* 5 (2014), 1–25.
- [62] Bernhard Hommel, Lorenza S Colzato, Rico Fischer, and Ingrid K Christoffels. 2011. Bilingualism and creativity: Benefits in convergent thinking come with losses in divergent thinking. *Frontiers in psychology* 2 (2011), 273.
- [63] Shang-Hsien Hsieh, Hsien-Tang Lin, Nai-Wen Chi, Kuang-Wu Chou, and Ken-Yu Lin. 2011. Enabling the development of base domain ontology through extraction of knowledge from engineering domain handbooks. *Advanced Engineering Informatics* 25, 2 (2011), 288–296.
- [64] Stacy Hsueh, Marianela Ciolfi Felice, Sarah Fdili Alaoui, and Wendy E Mackay. 2024. What Counts as 'Creative' Work? Articulating Four Epistemic Positions in Creativity-Oriented HCI Research. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–15.
- [65] Xi Hu, Yiwen Xing, Xudong Cai, Yihang Zhao, Michael Cook, Rita Borgo, and Timothy Neate. 2025. Designing Interactions with Generative AI for Art and Creativity: A Systematic Review and Taxonomy. In *Proceedings of the 2025 ACM Designing Interactive Systems Conference*. Association for Computing Machinery, New York, NY, USA, 1126–1155. doi:10.1145/3715336.3735843
- [66] Yanwei Huang, Wesley Hanwen Deng, Sijia Xiao, Motahhare Eslami, Jason I Hong, and Adam Perer. 2025. Viper: Towards systematic auditing of generative text-to-image models at scale. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–7.
- [67] Saskia Jaarsveld, Thomas Lachmann, and Cees Van Leeuwen. 2012. Creative reasoning across developmental levels: Convergence and divergence in problem creation. *Intelligence* 40, 2 (2012), 172–188.
- [68] Ivar Jacobson. 1999. *The unified software development process*. Pearson Education India, United States.
- [69] David G Jansson and Steven M Smith. 1991. Design fixation. *Design studies* 12, 1 (1991), 3–11.
- [70] Sanna Järvelä, Jonna Malmberg, Marta Sobocinski, and Paul A Kirschner. 2021. Metacognition in collaborative learning. In *International handbook of computer-supported collaborative learning*. Springer, Cham, 281–294.
- [71] Ninad K Jog and Ben Shneiderman. 1995. Starfield visualization with interactive smooth zooming. In *Working Conference on Visual Database Systems*. Springer, Chapman & Hall, Ltd., GBR, 3–14.
- [72] Santhosh John, Nazaraf Shah, and Craig Stewart. 2018. Towards a software centric approach for ontology development: Novel methodology and its application. In *2018 IEEE 15th International Conference on e-Business Engineering (ICEBE)*. IEEE, IEEE Computer Society, USA, 139–146.
- [73] C Maria Keet and Agnieszka Ławrynowicz. 2016. Test-driven development of ontologies. In *European Semantic Web Conference*. Springer, Springer-Verlag, Berlin, Heidelberg, 642–657.
- [74] C Maria Keet, Zola Mahlaza, and Mary-Jane Antia. 2019. CLaRO: a controlled language for authoring competency questions. In *Research Conference on Metadata and Semantics Research*. Springer, Springer International Publishing, Cham, 3–15.
- [75] Elizabeth A Kemp. 1996. The role of the individual project in teaching knowledge acquisition. In *Proceedings 1996 International Conference Software Engineering: Education and Practice*. IEEE, IEEE Computer Society, USA, 138–143.

- [76] Elisa F Kendall and Deborah L McGuinness. 2019. *Ontology engineering*. Morgan & Claypool Publishers, Cham.
- [77] Taewook Kim, Matthew Kay, Yuqian Sun, Melissa Roemmele, Max Kreminski, and John Joon Young Chung. 2025. Scaffolding Recursive Divergence and Convergence in Story Ideation. *arXiv preprint arXiv:2507.03307 N/A* (2025), 17 pages.
- [78] Tobias Kuhn. 2014. A survey and classification of controlled natural languages. *Computational linguistics* 40, 1 (2014), 121–170.
- [79] Jinhyuk Lee, Feiyang Chen, Sahil Dua, Daniel Cer, Madhuri Shanbhogue, Iftexhar Naim, Gustavo Hernández Ábrego, Zhe Li, Kaifeng Chen, Henrique Schechter Vera, et al. 2025. Gemini embedding: Generalizable embeddings from gemini. *arXiv preprint arXiv:2503.07891 N/A* (2025), 19 pages.
- [80] Kaihong Liu, WW Chapman, G Savova, CG Chute, N Sioutos, and Rebecca S Crowley. 2011. Effectiveness of lexico-syntactic pattern matching for ontology enrichment with clinical documents. *Methods of information in medicine* 50, 05 (2011), 397–407.
- [81] Yu Lu Liu, Wesley Hanwen Deng, Michelle S Lam, Motahhare Eslami, Juho Kim, Q Vera Liao, Wei Xu, Jekaterina Novikova, and Ziang Xiao. 2025. Human-centered evaluation and auditing of language models. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–7.
- [82] Paisley Livingston. 2009. Poincaré’s “Delicate Sieve”: On Creativity in the Arts. In *The Idea of Creativity*, Paisley Livingston (Ed.). Brill, Leiden, Netherlands, 129–146.
- [83] James Derek Lomas, Mihovil Karac, and Mathieu Gielen. 2021. 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.
- [84] Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica* 22, 3 (2012), 276–282.
- [85] Diane McKerlie and Allan MacLean. 1994. Reasoning with design rationale: practical experience with design space analysis. *Design Studies* 15, 2 (1994), 214–226.
- [86] Inc. Miro. 2025. *Miro - The Innovation Workspace*. Miro, Inc. <https://miro.com/> Accessed 2025-10-08.
- [87] Glaice Kelly Q Monfardini, Jordana S Salamon, and Monalessa P Barcellos. 2023. Use of competency questions in ontology engineering: A survey. In *International Conference on Conceptual Modeling*. Springer, Springer, Cham, Portugal, 45–64.
- [88] Jörg Müller, Daniel Polansky, Petr Novak, Christian Foltin, and Dimitry Polivaev. 2014. FreeMind - Free mind mapping software. https://freemind.sourceforge.io/wiki/index.php/Main_Page Accessed 2025-10-08.
- [89] Mark A Musen. 2015. The protégé project: a look back and a look forward. *AI matters* 1, 4 (2015), 4–12.
- [90] Robert Neches, Richard E Fikes, Tim Finin, Thomas Gruber, Ramesh Patil, Ted Senator, and William R Swartout. 1991. Enabling technology for knowledge sharing. *AI magazine* 12, 3 (1991), 36–36.
- [91] D. A. Norman. 1987. *Some observations on mental models*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 241–244.
- [92] Natalya F Noy, Deborah L McGuinness, et al. 2001. Ontology development 101: A guide to creating your first ontology.
- [93] Victor Ojewale, Ryan Steed, Briana Vecchione, Abeba Birhane, and Inioluwa Deborah Raji. 2025. Towards AI accountability infrastructure: Gaps and opportunities in AI audit tooling. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–29.
- [94] OpenAI. 2024. New embedding models and API updates. <https://openai.com/index/new-embedding-models-and-api-updates/> Accessed 2025-10-07.
- [95] Matthew J Page, Joanne E McKenzie, Patrick M Bossuyt, Isabelle Boutron, Tammy C Hoffmann, Cynthia D Mulrow, Larissa Shamseer, Jennifer M Tetzlaff, Elie A Akl, Sue E Brennan, et al. 2021. The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *bmj* 372 (2021), 9 pages.
- [96] Silvio Peroni. 2016. A simplified agile methodology for ontology development. In *International Experiences and Directions Workshop on OWL*. Springer, Springer-Verlag, Berlin, Heidelberg, 55–69.
- [97] María Poveda-Villalón, Alba Fernández-Izquierdo, Mariano Fernández-López, and Raúl García-Castro. 2022. LOT: An industrial oriented ontology engineering framework. *Engineering Applications of Artificial Intelligence* 111 (2022), 104755.
- [98] Valentina Presutti, Enrico Daga, Aldo Gangemi, and Eva Blomqvist. 2009. eXtreme design with content ontology design patterns. In *Proc. Workshop on Ontology Patterns*. CEUR-WS.org, Aachen, DEU, 83–97.
- [99] Hua Xuan Qin, Shan Jin, Ze Gao, Mingming Fan, and Pan Hui. 2024. 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*. Association for Computing Machinery, New York, NY, USA, 1–19.
- [100] Lila Rao, Han Reichgelt, and Kwaku-Muata Osei-Bryson. 2008. Knowledge elicitation techniques for deriving competency questions for ontologies. In *International Conference on Enterprise Information Systems*, Vol. 3. SCITEPRESS, SciTePress, Barcelona, Spain, 105–110.
- [101] Youssra Rebboud, Lionel Tailhardat, Pasquale Lisena, and Raphaël Troncy. 2024. Can LLMs Generate Competency Questions?. In *European Semantic Web Conference*. Springer, Springer Nature Switzerland, Cham, 71–80.
- [102] Yuan Ren, Artemis Parvizi, Chris Mellish, Jeff Z Pan, Kees Van Deemter, and Robert Stevens. 2014. Towards competency question-driven ontology authoring. In *European Semantic Web Conference*. Springer, Springer International Publishing, Cham, 752–767.
- [103] Gery W Ryan, H Russell Bernard, et al. 2000. Data management and analysis methods. *Handbook of qualitative research* 2, 1 (2000), 769–802.
- [104] Manojit Sarkar and Marc H Brown. 1994. Graphical fisheye views. *Commun. ACM* 37, 12 (1994), 73–83.
- [105] Juan F Sequeda, Willard J Briggs, Daniel P Miranker, and Wayne P Heideman. 2019. A pay-as-you-go methodology to design and build enterprise knowledge graphs from relational databases. In *International semantic web conference*. Springer, Springer-Verlag, Berlin, Heidelberg, 526–545.
- [106] Nigel Shadbolt and A Mike Burton. 1989. The empirical study of knowledge elicitation techniques. *ACM SIGART Bulletin N/A*, 108 (1989), 15–18.
- [107] Mary Shaw. 2011. The role of design spaces. *IEEE software* 29, 1 (2011), 46–50.
- [108] Ben Shneiderman. 2003. The eyes have it: A task by data type taxonomy for information visualizations. In *The craft of information visualization*. IEEE Computer Society, USA, 364–371.
- [109] Dean Keith Simonton. 2015. On praising convergent thinking: Creativity as blind variation and selective retention. *Creativity Research Journal* 27, 3 (2015), 262–270.
- [110] Lyle M Spencer and Phd Signe M Spencer. 1993. *Competence at Work models for superior performance*. John Wiley & Sons, USA.
- [111] Daniele Spoladore and Elena Pessot. 2022. An evaluation of agile ontology engineering methodologies for the digital transformation of companies. *Computers in Industry* 140 (2022), 103690.
- [112] Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez, and Mariano Fernandez-Lopez. 2015. The NeOn Methodology framework: A scenario-based methodology for ontology development. *Applied ontology* 10, 2 (2015), 107–145.
- [113] Sangho Suh, Meng Chen, Bryan Min, Toby Jia-Jun Li, and Haijun Xia. 2024. 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*. Association for Computing Machinery, New York, NY, USA, 1–26.
- [114] Sangho Suh, Bryan Min, Srishti Palani, and Haijun Xia. 2023. Sensecape: Enabling multilevel exploration and sensemaking with large language models. In *Proceedings of the 36th annual ACM symposium on user interface software and technology*. Association for Computing Machinery, New York, NY, USA, 1–18.
- [115] Jenifer Tidwell. 2010. *Designing interfaces: Patterns for effective interaction design*. O’Reilly Media, Inc., USA.
- [116] José Leomar Todesco, Sandro Rautenberg, Rafael Speroni, Ricardo Guembarovski, Fernando Ostuni Gauthier, et al. 2009. ontoKEM: A Web Tool for Ontologies’ Construction and Documentation.. In *IKE*. Springer Nature, USA, 86–92.
- [117] Inc. TopQuadrant. 2023. *TopBraid EDG 7.5 Documentation*. TopQuadrant, Inc. <https://www.topquadrant.com/doc/7.5/index.html> Accessed 2025-10-08.
- [118] Mike Uschold and Michael Gruninger. 1996. Ontologies: Principles, methods and applications. *The knowledge engineering review* 11, 2 (1996), 93–136.
- [119] Maarten W Van Someren, Yvonne F Barnard, Jacobijn AC Sandberg, et al. 1994. The think aloud method: a practical approach to modelling cognitive processes. *London: AcademicPress* 11, 6 (1994), 219 pages.
- [120] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017), 11 pages.
- [121] Markel Vigo, Samantha Bail, Caroline Jay, and Robert Stevens. 2014. Overcoming the pitfalls of ontology authoring: Strategies and implications for tool design. *International Journal of Human-Computer Studies* 72, 12 (2014), 835–845.
- [122] Denny Vrandečić, Lydia Pintscher, and Markus Krötzsch. 2023. Wikidata: The making of. In *Companion Proceedings of the ACM Web Conference 2023*. ACM, Austin, 615–624.
- [123] Bo Westerlund. 2005. Design space conceptual tool-grasping the design process. *Nordes N/A*, 1 (2005), 7 pages.
- [124] Mikael Wiberg and Erik Stolterman. 2014. 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*. ACM, Finland, 531–540.
- [125] Robin H Willemsen, Isabelle C de Vink, Evelyn H Kroesbergen, and Ard W Lazonder. 2023. The role of creative thinking in children’s scientific reasoning. *Thinking Skills and Creativity* 49 (2023), 101375.
- [126] Christopher Williamson and Ben Shneiderman. 1992. 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*. ACM, Denmark, 338–346.

- [127] Dawid Wiśniewski, Jędrzej Potoniec, Agnieszka Ławrynowicz, and C Maria Keet. 2019. Analysis of ontology competency questions and their formalizations in SPARQL-OWL. *Journal of Web Semantics* 59 (2019), 100534.
- [128] Robert J Youmans and Thomaz Arciszewski. 2014. Design fixation: Classifications and modern methods of prevention. *AI EDAM* 28, 2 (2014), 129–137.
- [129] J Diego Zamfirescu-Pereira, Richmond Y Wong, Bjoern Hartmann, and Qian Yang. 2023. Why Johnny can't prompt: how non-AI experts try (and fail) to design LLM prompts. In *Proceedings of the 2023 CHI conference on human factors in computing systems*. ACM, Germany, 1–21.
- [130] Bohui Zhang, Valentina Anita Carriero, Katrin Schreiberhuber, Stefani Tsaneva, Lucia Sánchez González, Jongmo Kim, and Jacopo de Berardinis. 2024. Ontochat: a framework for conversational ontology engineering using language models. In *European Semantic Web Conference*. Springer, Springer-Verlag, Greece, 102–121.
- [131] Bohui Zhang, Elisavet Koutsiana, Yihang Zhao, Albert Meroño-Peñuela, and Elena Simperl. 2025. Trustworthy Knowledge Graphs: Practices and Approaches. In *Handbook on Neurosymbolic AI and Knowledge Graphs*. IOS Press, Amsterdam, 363–384. doi:10.3233/FAIA250215
- [132] Weitao Zhang, Zsuzsika Sjoerds, and Bernhard Hommel. 2020. Metacontrol of human creativity: The neurocognitive mechanisms of convergent and divergent thinking. *NeuroImage* 210 (2020), 116572.
- [133] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. 2025. Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models. *arXiv preprint arXiv:2506.05176* N/A (2025), 14 pages.
- [134] Yihang Zhao. 2025. Leveraging Large Language Models for Ontology Requirements Engineering. In *Extended Semantic Web Conference ESWC*. Springer Cham, Slovenia, 11 pages. doi:10.1007/978-3-031-99554-5_40
- [135] Yihang Zhao, Xi Hu, Timothy Neate, Albert Meroño Peñuela, and Elena Simperl. 2025. A Survey on Interaction Design with Large Language Models for Ontology Requirements Elicitation with Competency Questions. *N/A* N/A (2025), 24 pages.
- [136] Yihang Zhao, Albert Meroño Peñuela, and Elena Simperl. 2024. User Experience In Dataset Search. In *International Conference on Computer-Human Interaction Research and Applications*. Springer, Springer Nature, Portugal, 113–130. doi:10.1007/978-3-031-83845-3_7
- [137] Yihang Zhao, Neil Vetter, and Kaveh Aryan. 2024. Using Large Language Models for OntoClean-based Ontology Refinement. *arXiv:2403.15864* [cs.AI] <https://arxiv.org/abs/2403.15864>
- [138] Yihang Zhao, Bohui Zhang, Xi Hu, Shuyin Ouyang, Jongmo Kim, Nitisha Jain, Jacopo de Berardinis, Albert Meroño-Peñuela, and Elena Simperl. 2024. Improving Ontology Requirements Engineering with OntoChat and Participatory Prompting. In *Proceedings of the AAAI Symposium Series*. AAAI Press, USA, 253–257. doi:10.1609/aaais.v41i1.31799
- [139] Weili Zhu, Siyuan Shang, Weili Jiang, Meng Pei, and Yanjie Su. 2019. Convergent thinking moderates the relationship between divergent thinking and scientific creativity. *Creativity Research Journal* 31, 3 (2019), 320–328.

A Literature Search: Supplementary Details

A.1 Step 1: Identification

As shown in Fig. 4, we designed a broad search query to capture many relevant studies. Instead of limiting it to LLM-specific terms, we used umbrella terms such as *AI* and *machine learning* to reflect the modular role of LLMs in multimodal systems that integrate other models for tasks like image generation, music synthesis, or style transfer.

The search yielded 40,184 unique records across three databases. We first restricted results to journal and conference publications. We then applied venue-level filtering following Hu et al. [65]. For each venue, we sampled the top 12 papers by relevance (12 was the smallest return across venues) and manually assessed topical alignment. Venues were excluded if fewer than 2 of 12 papers (i.e., <10%) addressed LLM-based creative systems with interaction designs. This removed venues where broad terms (e.g., “AI,” “art”) mostly retrieved irrelevant work (e.g., hardware optimization) while retaining venues with even modest but genuine contributions. We restricted the search to *title*, *abstract*, and *keywords*, as broader fields yielded many false positives, such as state-of-the-art in the body text. After these refinements, 3,275 papers were retained for screening.

A.2 Step 2: Screening

We considered diverse publication types, such as short papers, workshop and symposium papers, demos, and full papers, because they often present emerging interaction designs. To ensure screening consistency, the first two authors independently reviewed a random 20% subset ($N = 655$) of titles and abstracts at the beginning. Inter-rater reliability (Cohen’s kappa coefficient ($\kappa = 0.91$) [84]) was 0.82, indicating “strong” agreement (0.80–0.90). Discrepancies were resolved through discussion, mainly over whether outputs of LLM-based systems constituted artistic expression or creative design (EC-5).

A.3 Step 3: Snowballing

We conducted three rounds of forward snowballing using Google Scholar’s “Cited by” lists and two rounds of backward snowballing examining reference lists, completed in 2025. This procedure yielded 8 additional papers after duplicate removal and screening.

A.4 Step 4: Analysis

We first derived high-level categories from existing taxonomies of UIs [24] and interaction techniques [27]. We then performed bottom-up open coding on a random subset of 25 papers (50% of the dataset), allowing new codes to emerge within and beyond these categories. For instance, while the top-down structure included “sliders” as an input type, open coding showed the need to distinguish “single-parameter sliders” from “interpolation sliders,” leading to new sub-codes. This hybrid strategy balanced theoretical grounding with attention to domain-specific patterns [25], so the codebook captured both established interaction patterns and LLM-specific ones.

The first two authors collaboratively developed the codebook and resolved disagreements, mainly about boundaries between second-level UI codes in the same category, through discussion until consensus. After finalizing the structure, they iteratively defined each code. For interaction techniques, they extracted operational descriptions from the papers (e.g., how users navigate, filter, or cluster artifacts), synthesized these into generic definitions, and extended them by analysing how LLMs can be integrated to support divergent or convergent thinking. For UIs, they followed a similar process, adding visual sketches that abstracted common patterns across papers to depict archetypes instead of only text. The first two authors refined all definitions until they agreed on their technical accuracy and applicability to LLM-based creative systems.

After finalizing the codebook, both authors independently applied it to the remaining 25 papers, recording each code as binary (1 = present, 0 = absent). They compared codes and computed inter-rater reliability using Cohen’s kappa coefficient [84], which averaged 0.85 (“strong” agreement, 0.80–0.90). Disagreements, typically about slider functionality, such as whether a slider adjusted one variable or blended multiple variables, were resolved by jointly reviewing the relevant sections and figures and agreeing on the final codes.

B User Evaluation: Supplementary Details

B.1 Recruitment Details

Recruitment was conducted over one month via the W3C Semantic Web public mailing list⁹, an active forum for ontology engineers and related OE practitioners. As OE is a highly specialised field with a relatively small expert community compared to areas like general AI, recruiting many suitable participants was difficult, but our sample size (N=15) is comparable to prior evaluations of ontology tools [130, 138].

Additionally, 2/15 participants were only “slightly familiar” with knowledge elicitation in OE, and 3/15 were only “slightly familiar” with using CQs for it. This may be because some ontology engineers focus on knowledge formalization with limited knowledge elicitation expertise, while others emphasize knowledge elicitation but use techniques such as user stories instead of CQs. This variation is valuable, as it shows how OntoScope can support a broad range of ontology engineers.

B.2 Pilot Study Details

Before the main sessions, we ran a pilot with 3 ontology engineers (1 academic, 2 industry; all PhD-level; none re-recruited). Using preliminary OntoScope versions, they completed the task, and their feedback led to: (1) Protocol: confirming 20 minutes as sufficient for think-aloud. (2) OntoScope parameters: reducing CQs per suggestion from 5 to 3 to avoid overload; increasing terminology suggestions from 1 to 3 to preserve variety; and removing maximum limits on dimensions, CQs, and terms to encourage exploration. (3) Usability: adding colour-coded frames to each cluster to indicate coverage density.

B.3 Detailed Procedure

B.3.1 Step 1: Set-up and demographic questionnaire (5 mins). All 15 sessions were conducted online via Microsoft Teams. After receiving signed consent and approval for screen and audio recording, participants completed a demographic questionnaire covering gender, age, education, professional setting, current or previous OE project domains, and self-reported familiarity with knowledge elicitation in OE.

B.3.2 Step 2: OntoScope tutorial (10 mins). Participants received a link to OntoScope, shared their screen, and granted the researcher remote control. The researcher introduced the interface and demonstrated its purpose and functionality with a practice task on scoping a *university* ontology, which participants then repeated. This ensured familiarity with the system and time for questions. The researcher followed a scripted protocol so all participants received the same tutorial and a consistent understanding of the system.

B.3.3 Step 3: Concurrent think-aloud (20 mins). Participants performed a concurrent think-aloud [119] to evaluate how OntoScope supports divergent and convergent thinking in ontology scoping. They used OntoScope to define boundaries for one familiar domain from the 20 in the WDV-CQ-HA dataset [29] (e.g., university, airport, mountains). This dataset provides CQs with expert-validated answers across 20 broadly accessible Wikidata domains and served as the benchmark for user-generated CQs and answers.

During the main session, the researcher asked participants: “As you use OntoScope, please say out loud anything that currently helps or hinders you in coming up with relevant dimension values, CQs, or terms.” An observation checklist captured interface- or feature-related questions as supplementary usability evidence. Although usability was not the primary focus, issues were noted whenever participants reported features that hindered their divergent or convergent processes. These issues were then probed in the retrospective think-aloud, where participants were asked which interface or features had limited their process and why.

B.3.4 Step 4: Post-task questionnaire and retrospective think-aloud (10 mins). To capture overall perceptions (EQ1), participants completed a post-task Likert-scale questionnaire. Because OntoScope is intended to support both divergent and convergent thinking, we employed the Creativity Support Index (CSI) [28] as our measurement instrument (a validated scale for perceived creativity support) rather than general usability or workload measures such as the System Usability Scale (SUS) [26] or the NASA Task Load Index (NASA-TLX) [60], which focus primarily on efficiency, usability, or workload rather than on creative processes. We adapted ten CSI statements to ontology scoping, covering Enjoyment, Exploration, Expressiveness, Immersion, and Results Worth Effort. From the initial set of six CSI factors and twelve statements, we excluded the *collaboration* factor along with its two associated items, as they are not relevant to our case. Responses used a 10-point scale, and we computed scores by adapting the original CSI weighting method [28] to our case. This study concluded with a retrospective think-aloud [119], in which participants answered open-ended questions about which OntoScope interfaces, features, or dimension values supported or hindered their divergent and convergent thinking during boundary auditing, and why.

B.4 Embedding Model Selection

We chose *gemini-embedding-001* [79] because, on a randomly drawn 30% subsample, its outputs corresponded more closely to human assessments than those from *Qwen3-Embedding-8B* [133] and *text-embedding-3-large* [94]. All matches were reviewed by hand, since terms that are semantically similar are not necessarily ontologically identical (e.g., *composer* vs. *musician*), and several proposed matches were discarded.