

Identifying Needs in Patent Examiner Workflows Through Contextual Inquiry

Shanaya Malik

shanaya_malik@berkeley.edu
University of California, Berkeley
Berkeley, California, USA

Kelvin Zhao

kelvinyz@berkeley.edu
University of California, Berkeley
Berkeley, California, USA

Abstract

Patent examiners in technical domains must interpret legal claim language, search prior art databases, and synthesize evidence to evaluate whether a claimed invention is novel; yet, the tools available to them primarily support retrieval, not the interpretive and organizational work that follows. We conducted a contextual inquiry study with ten patent examiners working in areas such as cloud computing, video conferencing, and biomedical devices, observing their workflows in real time and analyzing the data using reflexive thematic analysis. We identified five recurring needs: (1) examiners lack systematic support for generating the synonyms required to bridge claim language and searchable terminology; (2) filing-date filtering tools are unreliable, leading to costly rework; (3) claim-to-evidence mapping is performed entirely through manual, improvised methods; (4) keyword hit locations within documents are opaque, producing false-positive relevance judgments; and (5) AI-assisted search tools do not support the iterative refinement strategies examiners rely on. Our findings suggest that future tool design should prioritize integration over automation, connecting disconnected workflow stages rather than replacing examiner judgment.

CCS Concepts

• **Human-centered computing** → **Empirical studies in HCI**; *HCI design and evaluation methods*.

Keywords

patent examination, prior art search, contextual inquiry, sensemaking, information work, HCI, need-finding

ACM Reference Format:

Shanaya Malik and Kelvin Zhao. 2026. Identifying Needs in Patent Examiner Workflows Through Contextual Inquiry. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 5 pages. <https://doi.org/XXXXXXX.XXXXXXX>

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

Conference acronym 'XX, Woodstock, NY

© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-XXXX-X/2018/06

<https://doi.org/XXXXXXX.XXXXXXX>

1 Motivation

Patent examination is one of the most consequential forms of technical auditing in the innovation economy; an examiner's determination of whether a claimed invention is novel directly shapes what technologies receive legal protection and how markets develop. For applications in complex technical domains—machine learning, integrated circuits, networked systems—this evaluation demands that the examiner interpret abstract legal claim language, translate it into concrete technical concepts, search across massive prior-art databases, and synthesize evidence to determine whether the claimed invention was already known.

The synthesis and evidence-mapping stages that follow the initial search are underexplored from an HCI perspective. Existing research has studied adjacent populations, including software developers navigating unfamiliar codebases [1], analysts constructing mental models from fragmented evidence [2], and professionals conducting code review at scale [3]. However, patent examining sits at the intersection of legal interpretation and technical comprehension, and this combination has not been a focus of need-finding research in the HCI or programming languages communities.

The tools that are currently available to examiners include databases such as the USPTO Patent Full-Text and Image Database (PatFT), the Patent Application Full-Text database (AppFT), and Google Patents, which are primarily designed for search and retrieval. These tools offer limited support for the interpretive work that follows, including decomposing claims into discrete elements, tracking which elements have been addressed by which references, and maintaining a coherent analysis across dozens of documents. This work is currently performed through ad hoc methods that vary among examiners and can include spreadsheets, margin notes, and mental bookkeeping, rather than through tools designed for the task.

Understanding where and how these gaps create friction will allow us to inform the design of specialized programming environments or domain-specific tools that could structure the evidence-linking process.

In this paper, we make three contributions: (1) an empirical account of how technical patent examiners discover, filter, and synthesize prior art in practice; (2) an identification of concrete needs grounded in observed breakdowns; and (3) implications for tool design that support the full arc of prior art evaluation.

2 Related Work

2.1 Sensemaking and Information Foraging in Professional Technical Work

The cognitive demands of patent examination have close parallels in research on how professionals organize and interpret complex information. Pirolli and Card proposed that complex analytical work involves cycling between “a foraging loop, in which analysts gather potentially relevant information [and] a sensemaking loop, in which they structure and interpret what they have found” [2]. Patent examiners follow a similar cycle of searching databases for prior art, while organizing retrieved evidence against specific claim elements constitutes sensemaking. Russell et al. characterized sensemaking as the iterative construction of a representational framework, finding that much of the difficulty in complex information work lies not in locating data but in organizing it to support judgment [4].

In software engineering, developers navigating unfamiliar codebases spend substantial time searching for scattered information, frequently losing track of reasoning chains. This challenge parallels examiners’ task of tracking how prior art relates to claim elements across many documents [1]. None of these studies, however, examines the intersection of legal interpretation and technical comprehension that defines patent examination.

2.2 Tools for Patent Search and Cross-Document Technical Analysis

In the patent domain, tool development has focused on the retrieval stage, as evidenced by the USPTO’s PatFT, AppFT, and the Cooperative Patent Classification (CPC) scheme; external tools like Google Patents and PATENTSCOPE offer semantic and cross-lingual search. Lupu et al. found that while recall-oriented search had improved substantially, helping users assess the relevance and coverage of results remained largely unaddressed [5]. AI-assisted tools are now available through public platforms such as IP.com and Google Patents, as well as the USPTO’s own internal systems, where examiners are required to run at least one AI-assisted search per application. These tools, however, still focus on retrieval rather than supporting the interpretive work that follows.

Outside of the patent domain, program comprehension tools help developers maintain context in large codebases. Storey et al. found that tool design often assumes a single cognitive strategy. In contrast, practitioners often alternate among top-down, bottom-up, and opportunistic approaches, which is relevant to patent examination [6]. Nersessian argued that practitioners’ mental model construction is central to expert judgment but is poorly supported by tools, and the persistent gap across these efforts is the lack of support for the mapping-oriented work that follows retrieval [7].

2.3 Contextual Inquiry as a Method for Studying Specialized Workflows

This study adopts contextual inquiry (CI), following the Master-Apprentice framework of Beyer and Holtzblatt, to surface knowledge and workarounds by observing practitioners in real work [8]. CI has been used successfully in HCI studies of specialized workflows, where Ko et al. [1] observed developers rather than simply interviewing them, and Lubin and Chasins [9] observed functional

programmers thinking aloud to surface patterns invisible to retrospective self-report. Zhang et al. used semi-structured interviews to surface unmet information needs among API designers [10], and our study follows these precedents by observing patent examiners in real time. For analysis, we use reflexive thematic analysis, allowing patterns to emerge from the data rather than imposing predetermined categories [11].

3 User Study Design and Procedure

The study was designed to address the research question: *What kinds of problems do technical patent examiners face during the iterative process of discovering, filtering, and synthesizing prior art to evaluate the novelty of complex technical claims?* This question was scoped to examiners in technical art units (such as software, biomedical engineering, or quantum computing), where the demands of translating between legal claim language and technical implementation are particularly demanding. We chose not to pre-specify hypotheses, since the study was exploratory, need-finding.

The participants were recruited through purposive and snowball sampling, beginning with two professional contacts who shared a recruitment message with colleagues and provided us with additional contacts. To supplement the remaining number of participants, we used cold messaging on LinkedIn to target individuals whose profiles indicated professional experience in technical patent examination. More specifically, the inclusion criteria required participants to be at least 18 years old, fluent in English, and currently or formerly employed in patent examination, with experience evaluating complex technical claims. We excluded individuals who exclusively examine non-technical patents and anyone unable to demonstrate their workflow using only publicly available documents.

Each session lasted approximately 45 minutes and was conducted remotely via Zoom. The sessions began with a brief unrecorded introduction of the participant and researcher, followed by an overview of the project scope and agenda for the session. The core of the session (~30–45 minutes, recorded) was a contextual inquiry following the Master-Apprentice model, in which the participant shared their screen and demonstrated their prior-art search and synthesis process for a given public patent while thinking aloud, with clarifying questions asked at natural pause points [8]. This was followed by a synthesis segment focused on how the participant maps and documents the relationship between prior art and specific claim elements, and concluded with a closing discussion addressing the most time-consuming and mentally demanding aspects of the workflow.

Each session produced three data artifacts stored on a password-protected Berkeley bDrive account: (1) a Zoom screen-and-audio recording capturing screen activity, search queries, and think-aloud narration; (2) typed session notes taken in real time; and (3) a post-session debrief document recording emerging patterns. The recordings were transcribed within Zoom itself and reviewed for accuracy. The participants were assigned coded identifiers (P01, P02, etc.), and we consolidated each participant’s transcript, session notes, and debrief into a single document as part of our input to our reflexive thematic analysis [11].

4 User Study Results

4.1 Analysis Process

We analyzed the consolidated session documents using reflexive thematic analysis following Braun and Clarke [11]. Each researcher independently conducted an initial round of open coding on their assigned participant's contextual inquiries, applying purely descriptive labels to segments that capture actions, reasoning, breakdowns, and workarounds within each patent examiner's workflow. Since our data include both think-aloud narration and question answers, as well as screen-recorded behavior, our coding explicitly distinguishes between what participants said and what actions they took, allowing us to capture both direct information from explanations and implicit information from instances where difficulties were not verbally articulated. As a result, many codes described domain-specific interactions, such as refining search queries, tracking claim coverage across references, and using specific internal tools to manage information across documents.

After independent coding, we compared our code sets across assigned participants and iteratively refined them through discussion, focusing first on resolving differences in interpretation and initial grouping, before collaboratively identifying higher-level patterns that showed recurring needs across multiple participants. We treated this stage as an iterative process, merging overlapping codes and preserving any meaningful differences reflecting underlying issues in the participants' workflows. We grouped related codes into candidate themes, tightened their definitions, and verified that each theme was supported by evidence from multiple participants. The five themes we present below represent the needs that emerged most consistently across our participant pool.

4.2 Findings

Need 1: Examiners lack systematic support for generating and managing the synonyms needed to translate legal claim language into effective search queries. A central challenge across sessions was the gap between claim terminology and the terminology used to retrieve relevant prior art; as P04 explained, the applicant “comes up with their own term, so we try to come up with terms that are going to be describing the same thing.” P06 described this as a problem of implied language: a term like *modder* in a medical device patent could return results about computer modification, and a term like *wearable* might not appear in a relevant document that uses a synonym, meaning those documents would be filtered out. Examiners relied on a patchwork of ad hoc resources, including a shared synonym crosswalk tool (P04), Google searches (P04), terminology from prior specifications, and personal experience. P06 described a case where an applicant's term for an ophthalmic structure appeared only in their own application, requiring consultation with a senior examiner to identify standard terminology. P07 noted that while examiners are taught about synonyms and Boolean connectors, applying this in unfamiliar art areas requires substantial independent research, alongside occasionally having to consult a senior examiner. Similarly, P10 described getting “stuck with the synonyms” when attempting to expand search queries in unfamiliar or highly technical applications, reinforcing that synonym generation remains an unsupported and cognitively

demanding step. Junior examiners were particularly disadvantaged, lacking the tacit vocabulary that senior examiners described as built through years of repetition (P01, P05, P06).

Need 2: Current search tools do not adequately support filtering by filing date, leading to wasted effort and rework.

Every participant referenced the effective filing date as a hard constraint on admissible prior art, yet existing tools provided limited support for enforcing it during search. P05 described completing an entire office action based on a reference that turned out to post-date the application's effective filing date, requiring a complete redo. P06 identified a related problem in which search systems frequently confuse publication dates with filing dates, and also confuse non-provisional filing dates with earlier provisional dates that establish actual priority. P06 called this “definitely a very good fix that can be done.” P01 noted that the rules for effective filing dates differ between U.S. and foreign applications, making even manual checks non-trivial. P08 described the filing-date constraint as deeply embedded in the workflow, requiring cross-referencing across domestic, provisional, and foreign filings, and P03 had initially demonstrated this in action, showing that of seven initial results, only one had a usable date (but determining this required a manual, time-consuming inspection of each).

Need 3: The transition from search to claim mapping relies on entirely manual, ad hoc methods with little integrated tool support.

After identifying promising prior art, examiners must document how specific passages address specific claim elements, which is a process called claim mapping. Every participant performed this using personal, improvised methods outside the search tool. P01 maintained a Word table; P03 copied claims as bullet points and mapped inline; P04 used a personal claim chart; and P05 copied claims into a Word document with parentheses after each element and drew dependency trees by hand. P06 described using pen and paper for claim trees and explained that the in-house software's mapping feature was so inaccurate that “pretty much no one at the USPTO” trusts it—errors in claim dependencies can distort the entire analysis. P07 described the process as lacking elegant tooling, stating that “it's a very manual process, [examiners] just start plopping it in there.” P08 described printing claims on paper and annotating them physically to manage cognitive load across simultaneous cases, and P05 explicitly identified this as a design opportunity, stating that integrating claim mapping into the search tool could save a significant amount of time and energy for examiners.

Need 4: Examiners struggle to locate and interpret relevant information within retrieved documents.

Examiners reported that tools display keyword hit counts without distinguishing whether hits appear in the specification (the detailed description most useful for mapping), the abstract, or the claims. P05 wanted hits specifically in the background and description sections, not in the abstract or claims, and shared that the ranking system orders results by total hit count regardless of location, forcing examiners to open documents and manually scan for relevant sections. P02 used a similar process of scrolling and combing through CTRL+F results of keywords to find relevant paragraphs. P01 expressed further frustration with granted patents, which require citing by column and line number rather than paragraph number, adding

manual overhead when recording where evidence appeared. P09 described that some internal tools support multi-highlighting to track keyword locations across claims, but public tools lack this capability entirely, forcing examiners to flip between separate documents to locate where relevant content appears manually. In practice, P10 also described relying on scanning titles, images, and highlighted keywords to triage relevance, further illustrating that examiners must manually piece together signals across document representations to determine whether a result is worth deeper inspection. Several participants also described needing to read beyond the keyword hit itself to determine whether it was actually meaningful in context, often revisiting multiple sections of the document to interpret how the term was being used. For example, P03 explained that when a result appears potentially relevant, they cannot rely on keyword matches alone and instead “have to go through the whole patent” to verify whether it actually aligns with the claim, since surface-level indicators like keyword proximity are often insufficient. Together, these patterns suggest that relevance assessment is slowed not only by the difficulty of locating keyword matches but also by the additional cognitive effort required to interpret their significance in complex technical documents.

Need 5: Existing AI-assisted search tools do not align with the iterative, refinement-based search strategy that examiners actually use. Participants with experience using AI or semantic search tools—including the USPTO’s similarity search, IP.com, and other platforms—consistently described them as inadequate. P02 emphasized that effective search “really involves appraising the art as you search for it, and refining your search based on that,” which current tools do not support. P06 described the similarity search as “hit or miss,” identifying the core problem as the tool’s inability to distinguish domain context—a medical-device search for “user interface” returns results from banking and unrelated software. P07, while noting IP.com “seems to interpret language a little bit more intuitively,” still gravitated toward non-patent sources like technical manuals as a more efficient strategy. P09 echoed the mixed assessment, noting that the similarity search “has mixed results—sometimes I just click that button... and it finds me 300 references, of which 2 or 3 are actually useful. Other times, it’s just completely off the mark.” P09 also described a workaround of feeding publicly available PG pubs into external AI tools to help devise search strategies, while noting that proprietary restrictions prevent using AI with actual application information. It appeared that, across most participants, AI tools were supplements rather than primary methods, reinforcing that future design should augment iterative workflows rather than replace them.

4.3 Implications for Future Research and Tool Design

Several of our findings align with patterns reported in prior work on professional information work, including the challenges of maintaining context across documents [1], organizing scattered evidence [4], and iterative refinement in expert search [2]. However, our findings also diverge from prior work in important ways. First, while Ko et al. [1] found that developers lose track of reasoning chains across files, patent examiners face an additional structural constraint: their reasoning must be organized element-by-element

against legally defined claim language, not simply against a mental model of the codebase. The failure mode is therefore not just lost context but legally indefensible analysis. Second, Lupu et al. [5] identified relevance assessment as an open problem in patent retrieval, but framed it primarily as a ranking and interface problem; our findings show that the deeper issue is the absence of integrated support for the interpretive work that follows retrieval—claim mapping, synonym management, and filing-date verification—which existing tools do not address at all. Third, Storey et al. [6] observed that developers alternate among cognitive strategies and argued that tools should accommodate this flexibility; we find a similar pattern among examiners, but with the added constraint that the output of their sensemaking must conform to a rigid legal structure, limiting how flexibly they can reorganize their analysis. Finally, while Pirolli and Card’s sensemaking model [2] describes foraging and sensemaking as iteratively coupled, our participants described AI-assisted search tools that treat these as separate, one-shot operations (Need 5), suggesting that current tool design actively works against the iterative workflow that prior theory predicts experts will use.

These differences suggest that future research should more explicitly account for domains where information work is not only interpretive but also constrained. Rather than treating search and sensemaking as loosely coupled activities, researchers should consider how formal requirements, such as legal standards or structured mappings, shape both the workflow and the types of errors that occur. This also suggests a need for studying how experts convey intermediate reasoning steps under such constraints, particularly in environments where traceability and justification are critical.

Our findings also suggest that future tool design could prioritize integration over automation. Rather than replacing examiner judgment with AI-driven search, the most impactful opportunities may lie in reducing manual overhead for tasks examiners already perform, such as synonym management, date filtering, and claim-to-evidence mapping. More broadly, this points to the importance of designing systems that better support iterative refinement workflows (Need 5) and connect currently disconnected stages of work into a unified environment, rather than optimizing individual components in isolation.

Acknowledgments

We thank Professor Sarah E. Chasins for her guidance throughout this project and the CS 294 course at UC Berkeley. We also thank the nine patent examiners who generously shared their time, expertise, and workflows with us.

References

- [1] A. J. Ko, B. A. Myers, M. J. Coblenz, and H. H. Aung. An Exploratory Study of How Developers Seek, Relate, and Collect Relevant Information during Software Maintenance Tasks. *IEEE Trans. Softw. Eng.*, 32(12):971–987, 2006.
- [2] P. Pirolli and S. Card. The Sensemaking Process and Leverage Points for Analyst Technology as Identified Through Cognitive Task Analysis. In *Proc. Int. Conf. Intelligence Analysis*, volume 5. MITRE, McLean, VA, 2005.
- [3] C. Sadowski, E. Söderberg, L. Church, M. Sipko, and A. Bacchelli. Modern Code Review: A Case Study at Google. In *Proc. ICSE-SEIP '18*, pages 181–190. ACM, 2018.
- [4] D. M. Russell, M. J. Stefik, P. Pirolli, and S. K. Card. The Cost Structure of Sensemaking. In *Proc. CHI '93*, pages 269–276. ACM, 1993.
- [5] M. Lupu, K. Mayer, N. Kando, and A. J. Trippie. *Current Challenges in Patent Information Retrieval*, volume 29 of *The Information Retrieval Series*. Springer,

- 2013.
- [6] M.-A. D. Storey, F. D. Fracchia, and H. A. Müller. Cognitive Design Elements to Support the Construction of a Mental Model during Software Exploration. *J. Systems and Software*, 44(3):171–185, 1999.
 - [7] N. J. Nersessian. *Creating Scientific Concepts*. MIT Press, 2008.
 - [8] H. Beyer and K. Holtzblatt. *Contextual Design: Defining Customer-Centered Systems*. Morgan Kaufmann, 1997.
 - [9] J. Lubin and S. E. Chasins. How Statically-Typed Functional Programmers Write Code. *Proc. ACM Program. Lang.*, 5(OOPSLA):Article 155, 2021.
 - [10] T. Zhang, B. Hartmann, M. Kim, and E. Paez. Enabling Data-Driven API Design with Community Usage Data: A Need-Finding Study. In *Proc. CHI '20*, pages 1–13. ACM, 2020.
 - [11] V. Braun and V. Clarke. Using Thematic Analysis in Psychology. *Qualitative Research in Psychology*, 3(2):77–101, 2006.