Exploring Fairy Cursor as a Form of AI Agent for In-the-Flow Assistance: Design Opportunities and Challenges

Yining Cao
rimacyn@ucsd.edu
University of California, San Diego
La Jolla, CA, USA

Jim Hollan
hollan@ucsd.edu
University of California, San Diego
La Jolla, CA, USA

Haijun Xia
haijunxia@ucsd.edu
University of California, San Diego
La Jolla, CA, USA

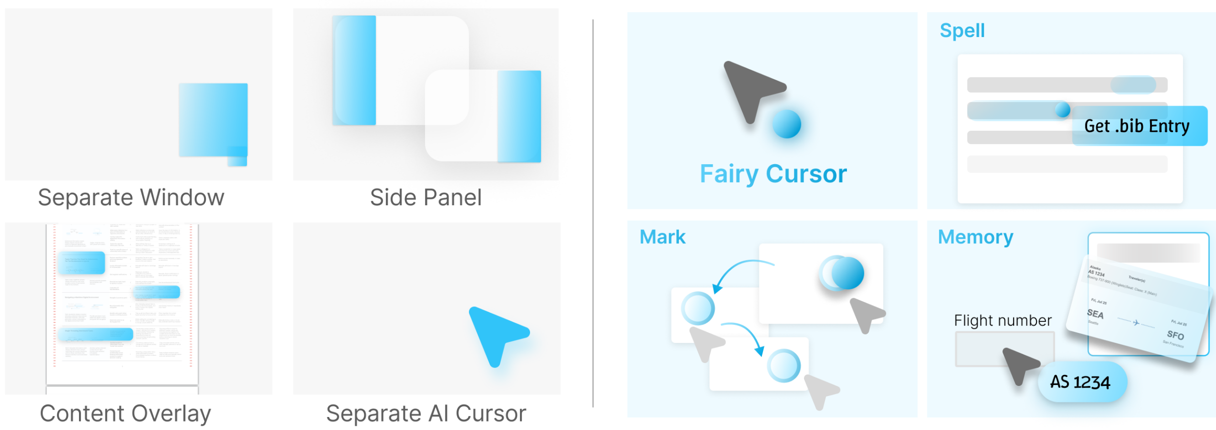


Figure 1: This paper explores cursor-following as a form factor for in-the-flow assistance. The left side summarizes four common forms of AI assistance in existing desktop environment: separate windows, side panels, content overlays, and standalone AI cursors. The right side introduces Fairy Cursor, a cursor-following agent used in this study as a design probe to explore the opportunities and challenges of cursor-centric assistance. Fairy Cursor operationalizes this form through three interaction primitives: Mark (anchoring and revisiting context), Spell (proactive, lightweight actions derived from user's past interaction patterns), and Memory (holding and resurfacing provisional information near the cursor).

Abstract

What would it mean for a digital assistant to stay with our cursor, rather than as a chatbot in a separate window? By staying near the user's actions, such assistance promises lightweight, continuous, in-the-flow help, but also introduces unique risks of intrusion. However, little is known about the tasks it is suited for, and the design considerations it entails. This study offers an initial investigation of this space through a multi-stage design inquiry. First, a retrospective think-aloud study with nine participants reveals common inefficiencies in everyday information work where cursor-centric support may be valuable. Building on these observations, we proposed the Fairy Cursor as a design probe and developed a proof-of-concept environment to examine users' interpretations, preferences and concerns around it. Our findings surfaced key challenges and opportunities in designing cursor-centric assistants, including how users tolerate presence, interpret initiatives, and balance assistance with ongoing engagement. We conclude with design implications for in-the-flow human–AI collaboration and outline directions for future systems.

DIS '26, Singapore, Singapore

ACM ISBN 979-8-4007-2563-0/2026/06
https://doi.org/10.1145/3800645.3813082

CCS Concepts

• **Human-centered computing → Graphical user interfaces**; *User interface design*.

Keywords

Cursor-centric assistance; Information Management; human–AI collaboration; Agent; Cursor

ACM Reference Format:
Yining Cao, Jim Hollan, and Haijun Xia. 2026. Exploring Fairy Cursor as a Form of AI Agent for In-the-Flow Assistance: Design Opportunities and Challenges. In *Designing Interactive Systems Conference (DIS '26), June 13–17, 2026, Singapore, Singapore.* ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3800645.3813082

1 Introduction

Most AI assistants on the desktop appear in a few familiar forms. Some live in separate windows, where users step away from their work to ask questions or delegate tasks [32]. Others appear as in-line helpers, such as grammar checkers or smart suggestions [23], embedded within specific documents or fields. More recently, autonomous agents have emerged, which can operate the interface itself, moving the cursor and manipulating the interface [1]. Each of these forms supports a different style of assistance. Chat-based agents are powerful for explicit requests, but pull users out of their workflow. Inline helpers offer quick, low-friction help, but are confined to narrow use cases. Autonomous agents can perform complex procedures, but often take over the interface, leaving little room for shared control or in-the-moment adjustment.

In this work, we explore the potential of a new form of assistive agent, which we call Fairy Cursor, that augments the mouse cursor with a lightweight following "Fairy." Our design is motivated by the observation that the cursor is not merely a pointing device, but one of the most continuous and embodied traces of a user's attention and intention in graphical interfaces. Nearly every interaction—selecting, navigating, manipulating, and referencing—passes through the cursor, making it a uniquely powerful locus at which to situate assistance. Unlike external agents that require users to shift attention or step away from their ongoing work, a cursor-centric agent can remain embedded within the flow of action. Additionally, by attaching a Fairy to the user's cursor—rather than granting AI control of the user's cursor or introducing an independent cursor—this form factor has the unique benefit of preserving users' agency while enabling expressive and controllable agentic behaviors.

However, it also introduces unique challenges. A cursor-following agent is continuously present and tightly coupled to the user's focus of attention, making it prone to distraction, intrusion, or misalignment with moment-to-moment intent. Unlike chat-based assistants that are explicitly invoked, a cursor-following agent must coexist with ongoing activity at all times, continuously negotiating when to act and when to remain silent.

To balance the distinctive advantages of cursor-following assistance with these design challenges, several questions must be addressed. Which tasks are best served by assistance that remains at the cursor? How do users interpret initiative from a companion that moves with them? What design qualities make such a presence feel helpful rather than irritating?

We investigate these questions through a multi-stage design inquiry. First, we conducted a retrospective think-aloud study in which nine participants recorded their everyday computer work and reflected on moments where in-the-flow assistance would have been valuable. From these interviews, we synthesize three core needs for in-the-flow assistance: (1) the need to refer back to prior context with lightweight and persistent anchors, (2) the need to accelerate small but repeated actions without committing to full automation, and (3) the need to hold ephemeral information without relying on heavyweight systems.

To further explore how might a cursor-following agent support these needs and elicit users' perceptions and challenges around cursor-centric assistance, we designed Fairy Cursor as a design probe built around three interaction primitives—Mark, Spell, and Memory—that operationalize referential anchoring, lightweight action reuse, and provisional working memory at the cursor. We intentionally abstracted away questions of model accuracy and system integration in order to isolate interaction-level concerns inherent to this form factor. Therefore, we instantiate Fairy Cursor in a web-based interactive sandbox to make otherwise abstract design questions tangible and discussable. Through a study using this playground, we examine how users experience a secondary cursor, interpret and appropriate its primitives, and negotiate helpfulness, control, and intrusiveness across different collaboration dynamics.

Our findings reveal both the promise and the challenges of cursor-centric assistance. Participants appreciated micro-task support that helped them maintain momentum and retain a sense of control when working with Fairy Cursor. We also identified several factors that are critical to the usefulness and usability of this interaction paradigm, including temporally appropriate intervention, clear communication of the assistant's state and intent, sensitivity to task context, and support for evolving personalization over time.

Together, our work contributes a grounded characterization of where in-the-flow assistance is needed, and surfaces design opportunities and challenges for cursor-centric AI agents. We conclude by outlining implications for in-the-flow human–AI collaboration and directions for future systems that aim to support micro-tasks, ephemeral information, and interleaved workflows rather than relying on full automation or explicit task delegation.

2 Related Work

This work investigates how a cursor-centric form can provide in-the-flow assistance during everyday information work. We draw on three lines of prior research that together characterize both the need for such assistance and relevant design approaches. We first review studies of fragmented information work to establish the nature of the problem space. We then examine existing forms of AI assistance and their task scopes, highlighting the gap addressed by our work. Finally, we situate our contribution within research on augmented and embodied cursors, motivating the cursor as a promising locus for situated, continuous assistance.

2.1 Fragmented, Frustrating, and Fragile Information Tasks

Despite tremendous advances in computing capacity, speed, and connectivity, the activities of today's information workers have become increasingly fragmented, frustrating, and fragile [5, 38]. A significant aspect of fragmentation can be attributed to the multitasking and distributed nature of human information activity [12, 21, 30, 52]. With information activities distributed across an increasing number of applications, services, and devices [7, 52], users often need to collect information from diverse sources and process it with multiple tools. Observational studies of office workers reveal that during a typical day, information workers spend an average of only 12 minutes on any given task, and most "uninterrupted events" average only 3 minutes in duration [38]. A variety of approaches have been developed to mitigate this problem, including advanced organizational systems for managing information activities [13, 47] and integrated environments designed for specific activities [8, 9].

This body of literature establishes a key premise for our work: many breakdowns arise not from a lack of information or automation, but from misalignment between available support and the moment-to-moment flow of activity. Rather than seeking to eliminate fragmentation as prior work has done, we take the position that fragmentation is an enduring characteristic of information work. Accordingly, our focus is on providing assistance that helps users traverse moments of fragmentation quickly, before they escalate into serious disruptions to flow.

2.2 Forms and Scopes of Intelligent Assistance

Leveraging machine intelligence and automation to assist with complex information and interface tasks has been a long-running thread of research in HCI [28, 36, 44]. Prior work has explored two complementary approaches, each reflecting a different balance between delegation and direct user engagement.

One approach embodies machine intelligence in the form of digital assistants to leverage familiar human social interaction skills and natural language input, with the goal of delegating aspects of complex tasks to an assistant. Assistants range from virtual humanized assistants as seen in Knowledge Navigator [4], proactive animated virtual characters such as Microsoft's Clippy [17], to modern natural language-based assistants such as Siri [32] and ChatGPT [40]. However, these types of assistants require users to pause, articulate their intent, and step away from ongoing work, and they often lack access to the rich, situated context needed to respond appropriately to open-ended or partially formed requests. As a result, these systems are best suited for discrete queries or well-scoped delegation, rather than for supporting the incremental, interleaved nature of everyday information work.

The other approach has been to integrate automation into direct manipulation interfaces as interactive automation or suggestions [27, 44], preserving the feeling of engagement with the content being manipulated [31]. In these systems, assistance is tightly coupled with the content: suggestions can be accepted opportunistically when they align with users' intentions, or dismissed with little cost when they do not. Common examples include automatic query suggestions and grammar checking in word processors. Recent HCI research has extended this approach to more complex tasks, such as those that help retrieve relevant information based on the current context [50, 53] and intelligently infer and adjust to user's evolving intentions from their interactions with the systems [14, 33, 46] This approach brings immediate added-values to users "while preserving their sense of control and their responsibility" [44]. However, since such assistance is typically embedded within specific applications, documents, or interaction contexts, its scope is often narrowly constrained within specific use cases. As a result, users must still manually bridge gaps between contexts, reconstruct prior states, or reapply actions elsewhere during interleaved work. These limitations echo broader findings that intelligent support often breaks down when tasks span multiple tools or evolve dynamically over time [48].

Our work explores a form of assistance that emphasizes proximity, partial action, and shared control. We focus on the cursor as a potential locus where automated services could be elegantly coupled with direct manipulation [28], while remaining capable of transforming into a dedicated, embodied assistant when users seek deeper automation.

2.3 Evolving, Augmented, Embodied Cursors

Since its debut in the historical Mother of All Demos [18], the graphical pointer—cursor, has become one of the most important interface elements for graphical user interfaces (GUIs). The shape and form of a cursor have been evolving for the past 50 years, from a simple vertical line controlled by a mouse to one that can take various forms (e.g., arrow, hand, hourglass, dot) depending on the content it is pointing at or the modality controlling it.

The cursor is used for most, if not all GUI interaction, such as pointing, selecting, manipulating graphical objects, and interacting with interface widgets. Research has explored how to enhance interaction with the mouse cursor, such as adding additional control degrees of freedom with novel pointing devices [6, 25], employing dynamic activation area to enhance target acquisition [24], and leveraging multiple cursors to reduce movement time towards desired targets [51].

The cursor can also be seen as an avatar of its operator in the digital space. Cursor movement can be controlled by mouse or gaze, and both can be used to infer the users' intention [15, 29]. In collaborative editing and design applications, multiple users are often represented with cursors of different colors to improve awareness and allow identification [19, 22]. Recently, AI advances have enabled the exploration of granting AI agents to operate the cursor. AI agents can receive users' natural language commands and then control the cursor [3] or browser [42], or as a separate autonomous cursor to complete users' tasks [1].

Granting AI direct control over the user's cursor, however, turns the cursor into a site of full task delegation, undermining the distinctive collaborative affordances that multi-cursor interaction can bring to digital work. Instead, we aim to preserve users' agency over their own cursor while providing a balanced degree of control to the AI agent. Our design exploration with Fairy Cursor investigates this alternative by introducing a second, assistive cursor that remains near the user's own, enabling lightweight, negotiable collaboration directly at the locus of action.

3 Study 1: Understanding the Needs for In-the-Flow Assistance

This study explores opportunities for cursor-centric assistance that help users remain in the flow of everyday information work. Central to this goal is identifying moments of inefficiency rooted in authentic user behavior. To this end, we conducted a retrospective think-aloud study guided by the following research questions:

- What types of inefficiencies disrupt users' flow during everyday information tasks?
- What in-the-flow assistance needs do users have that are better served by a cursor-centric agent than by full task delegation?

3.1 Participants

We recruited 9 participants (mean age = 27) who regularly engage in information computer tasks (6 reported 7-9 hours daily; 3 reported 10+ hours daily). Participants were sourced through public communication channels including email and institutional Slack workspaces. All participants provided informed consent and received a $15 Amazon gift card as compensation.

3.2 Study Procedure

The study consisted of two sessions: a self-recorded session of participants' natural computer use, followed by a retrospective think-aloud session guided by the first author.

3.2.1 *Task Recording.*

Participants independently recorded at least 30 minutes of typical information work on their personal devices. We emphasized that the tasks should be authentic and representative of their natural workflows. To address privacy concerns, participants were not required to share the recording footage with the researchers.

3.2.2 *Retrospective Think-Aloud Session.*

Each session began with a brief motivation of the Fairy Cursor concept, accompanied by a demonstration of an early prototype. This was intended to prime participants to identify opportunities where this form of assistance could be beneficial. Then participants were instructed to play back their screen recordings while verbalizing their actions and thought processes. We reviewed the recordings in detail to understand the context and flow of tasks. In this process, we adjusted the playback speed as needed, and avoided skipping sections except when they contained private information or when participants were not actively using their computers.

To facilitate discussions around potential assistance, throughout the session, the researcher prompted participants to identify: (1) moments of friction, inefficiency, or frustration, and (2) situations where assistance could have improved their experience. For each identified instance, we asked participants to describe what kind of help a fingertip assistant could ideally provide in that moment.

Table 1: Details of the Recorded Working Sessions.

PID	Duration	Tasks Performed
P1	30 min.	Grading class assignment submissions
	42 min.	Updating Overleaf template
P2	77 min.	Prototyping in Figma → reading email → writing → video conferencing
P3	62 min.	Programming
P4	31 min.	Learning a video tutorial while following on Unity
	37 min.	Rendering and testing Unity program
P5	29 min.	Figma Prototyping → exploring travel destination → sending messages → learning about programming knowledge → Figma Prototyping
P6	31 min.	Reviewing dataset → collaborative writing
	40 min.	Grading → replying to Slack → managing flight information
P7	67 min.	Replying to Slack → replying to email → replying to student request → replying to email
P8	40 min.	Writing a draft of a research article
P9	65 min.	Uploading a Manuscript to Arxiv → replying to Slack message → filling a reimbursement form

3.3 Results and Analysis

In total, we analyzed 12 working sessions recorded by 9 participants, covering a diverse range of tasks including writing, programming, travel planning, email composition, and more (see Table 1 for details). While we only had access to screen recordings shared by the participants during the think-aloud sessions, all sessions were audio-recorded and transcribed for analysis, focusing on moments when participants experienced friction, hesitation, or breakdown—where their attention momentarily drifted from the tasks at hand.

From our analysis, we identified three recurring classes of flow disruptions, each manifested through multiple concrete micro-tasks. These disruptions revealed three needs for in-the-flow assistance: (1) referring back to prior context with lightweight and persistent anchors, (2) accelerating small but repeated actions without committing to full automation, and (3) holding provisional, ephemeral information without relying on heavyweight systems. Below, we describe the empirical findings around these three categories.

4 Opportunities for Cursor-Centric, In-the-Flow Assistance

We present our findings on three common disruptions to users' flow, along with the types of in-the-flow assistance participants need. For each one, we summarize: (1) the underlying problem, (2) why it persists even with access to LLMs, and (3) how in-the-flow support can address it. Table 2 illustrates these findings with the corresponding concrete tasks derived from our interviews.

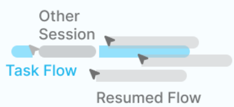
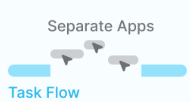
4.1 Referring Back to Prior Context with Lightweight Anchors

The multi-tasking nature of real-world activity means we are frequently doing two things: starting new tasks and resuming old ones. However, unlike physical workspaces, where materials naturally accumulate and provide contextual cues, our digital environments offer little built-in support for helping us (re)orient ourselves.

These breakdowns can appear in small moments—such as losing one's place in a codebase while scrolling to find a variable name (P3) or forgetting the navigation path after following a chain of hyperlinks (P2). It can be even more painful in larger transitions, such as returning to a task after a pause or switching between loosely related activities. Participants described opening their computer to a sea of tabs and windows with little indication of which ones mattered now (P5, P7). Another common friction occurs during "gap times," such as waiting for a process to complete or taking a brief break. Participants described wanting to squeeze small, productive tasks into these windows but often found themselves distracted longer than intended or struggling to retrieve a task that fit the moment from past activities. As P4 noted, "*It would be great if the system knew I had small to-dos in my notes and suggested one based on how long I'll be waiting on this rendering*." Without a global mechanism for preserving task state, users rely on scattered screenshots, timestamped notes, or their own memory to reconstruct prior work.

This motivates lightweight, situated traces—ways to leave anchors in context which allow assistance to help users refer back to prior context where it originally occurred, supporting navigation, resumption, and orientation.

Table 2: Observed Workflow Inefficiencies and Task-level Opportunities for In-the-Flow Assistance.

Disrupted Flow in Current Experience	Desired In-the-Flow Assistance	Task Needs In-the-Flow Assistance	# Cases Mentioned	Representative Scenario	Current User Tactics
Referring Back to Prior Context with Lightweight Anchors  Resuming a task often requires retracing scattered cues across disconnected materials, with little persistent context preserved in the digital environment itself.	 Provide lightweight anchors and persistent traces within the digital context to support navigation and resumption.	[N1.1] Navigate back to a previous digital artifact	7	After taking a screenshot, want to go to a specific window to insert it, but the window is buried within tabs [P2]	Manually search through or adjust window to keep it visible on the side
		[N1.2] Recontextualize after navigation	4	After following several links to locate a referenced paper, lose track of where the original reading point was left [P6]	Use browser history to find or repeatedly click "back"
		[N1.3] Identify entry point when resuming a working session	3	Tidy up all the leftover tabs and figure out what to focus on first [P5, P7]	Reconstruct current progress from memory
		[N1.4] Point to a specific piece of information with context	3	Refer a colleague to a particular paragraph in a PDF during a Slack discussion [P7]	Take a screenshot, or copy and paste text from the source with an explanatory note / annotation
		[N1.5] Receive customized updates on an ongoing process	3	Want to receive a notification only when a specific person messages within a certain time window [P5]	Manually check notifications or adjust global/system settings
		[N1.6] Determine what to do during gap time	2	While waiting for a rendering to finish, consider doing a small leftover task but unsure what fits the time window [P4]	Manually browse notes or to-do lists; estimate task fit by intuition
Accelerating Small but Repeated Actions Without Full Automation  Small, repetitive actions often feel ad hoc and rarely justify the overhead of fully formalized task automation.	 Actively surface reusable functionalities based observed interaction patterns.	[N2.1] Cross-reference multiple information sources to compare and synthesize for decision-making	7	Browse a restaurant's menu while synthesizing reviews from multiple sources to inform food choices [P5]	Keep multiple windows open/ side-by-side and manually switch, scan and align information across them
		[N2.2] Quickly transform information from one format into another	7	Convert citations from links in Google Doc into BibTeX entries in Overleaf [P8]	Copy and paste entries one by one; manually reformat or use ad hoc content editing tools
		[N2.3] Perform repetitive actions over long operation distance	5	When grading, Repeat the actions: giving feedback, submitting a grade, then clicking the next button [P1]	Perform action manually; or rely on software extensions
		[N2.4] Locate information buried in UI hierarchies	4	Find and copy .bib citation from Google Scholar page [P9]	Manually drill down; leverage search; or rely on third-party tools
		[N2.5] Execute low-level, physically hard-to-perform actions	2	Select multiple paragraphs that overflow the screen [P2]	Manually sustain the selection gesture while scrolling, repositioning
Holding Ephemeral Information Without Relying on Heavyweight Applications  Ephemeral information is forced to be pushed into long-term storage, breaking flow and forcing premature organizational decisions.	 Provide a simple dedicated space that holds and resurfaces what matters in the moment, without requiring long-term commitment.	[N3.1] Take a quick note/ reminder for the future self with context	11	While grading, receive a Slack notification linking to a paper to read later [P6]	Switch to a note-taking app or create a note elsewhere while manually documenting the context
		[N3.2] Temporarily pin information for reuse in another context	6	Make reference to internship start/end dates in an email to fill out forms [P2]	Save the piece of information in an unsorted document, physical note; or rely on working memory
		[N3.3] Have a place for ephemeral information editing	6	Draft an email while experimenting with multiple phrasings and tones before finalizing [P7]	Open a separate editor and create duplicate drafts
		[N3.4] Search for information mid-task	5	While writing, look up a definition or verify a citation [P8]	Browse or ask an AI assistance in a separate window

4.2 Accelerating Small but Repeated Actions without Full Automation

A second class of breakdowns consists of tasks that are plainly repetitive or tedious, yet too minor or idiosyncratic to justify full automation. These actions are typically embedded within larger activities—such as navigating interfaces, transforming information, or coordinating content across multiple windows. Although the cost of any single action is negligible, their accumulation produces a pervasive sense of friction and drudgery.

For example, one participant described repeatedly navigating through multiple screens to grade assignments, enter feedback, and return to a submission list (P1). Although the sequence was performed repetitively, it was described as "*annoying but not worth automating*." Similar patterns appeared when participants navigated nested interfaces to locate information (P9), adjusted settings via cumbersome menus (P2), or performed low-level interactions such as selecting and scrolling through overflowing text (P2).

This friction can appear frequently in cross-referencing tasks, such as converting citations between formats (P8), copying structured information across applications or visually searching related content between windows (P7). These actions were repetitive and effortful, yet tightly coupled to local context, making them ill-suited to existing automation tools (e.g., scripts or editable workflows). For scenarios where automation could be possible, participants reported that they would have to formalize intent, configure rules, or relinquish control. The costs of these outweighed the immediate burden of performing the action manually (P1, P2, P5, P8, P9).

These findings motivate in-the-flow assistance that speeds up small, repeated actions through lightweight, reusable mechanisms embedded directly in ongoing interaction. Such support should reduce physical and navigational burden while allowing users to stay engaged, retain control, and adjust actions in the moment.

4.3 Holding Ephemeral Information with Lightweight In-Situ Storage

Participants frequently encountered moments in which they wanted to temporarily retain information: a paper to read later (P6), a date needed to complete a form (P8), an alternative phrase while drafting an email (P7), or a piece of information encountered that they wanted to share with a colleague (P9). Although many tools exist for notes, bookmarks, and reminders, participants often chose not to use them in these situations. Deciding where to store the information—and shifting context to do so—felt disproportionate to its anticipated value and short-lived relevance.

Instead, participants relied on fragile, ad hoc strategies: keeping browser tabs open indefinitely, duplicating drafts, taking screenshots, or holding details in working memory. These tactics preserved momentum in the moment, but cluttered the workspace and increased the effort required to retrieve or make sense of the information later.

This sense of "heaviness" also appeared in interactions with AI tools. Participants described wanting to make brief, context-bound queries—such as clarifying a detail or rephrasing a sentence—but hesitated to do so because it required opening a separate chat window and re-establishing context. While temporary chat modes reduced persistence, they did not address the core issue: the interaction remained detached from the on-screen content that gave the question meaning. As a result, participants described these interactions as disruptive to their flow, undermining the spontaneity that made the information need ephemeral in the first place.

Across these cases, the challenge was not a lack of storage options, but the absence of support for provisional information—content that users want to capture, hold nearby, and possibly discard, without committing to organization or long-term retention.

These observations motivate the need for in-the-flow assistance that supports short-lived, context-bound information by allowing users to capture, resurface, and release it as needed, without forcing explicit storage decisions or interrupting ongoing activity.

4.4 Summary

Taken together, the findings from this study clarify both where users experience breakdowns in flow and what kinds of assistance they find appropriate in those moments. We found that inefficiencies that disrupt flow during everyday information work are rarely dramatic failures or missing capabilities. Instead, they appear as small, cumulative frictions embedded within ongoing activity—such as losing track of prior context, repeatedly performing low-level actions, or holding provisional information without a place to put it. These moments often go unnoticed while they occur, yet accumulate into sustained cognitive and interaction overhead.

The opportunities for cursor-centric assistance we observed are not defined by complete tasks, but by micro-moments within tasks—points where attention, memory, or motor effort briefly breaks down. By synthesizing these observations, we arrive at three core scenarios for in-the-flow assistance: supporting reference to prior context, enabling lightweight reuse of small actions, and holding ephemeral information near ongoing activity. These scenarios directly informed the design of Fairy Cursor and its core interaction primitives, which we introduce next.

5 Interaction Concepts of Fairy Cursor

The retrospective think-aloud study surfaced multiple opportunities for in-the-flow assistance, and raised questions about how such assistance should be experienced in practice. Specifically, how users interpret an assistive presence at the cursor, how they negotiate initiative and control, and what kinds of interaction feel supportive rather than distracting. To explore these questions, we proposed the following interaction concepts of the Fairy Cursor.

Fairy Cursor is envisioned as a cursor-following agent that stays near the user's primary cursor and offers assistance directly at the locus of interaction. Its design consists of three interaction primitives, Mark, Spell and Memory, informed by the opportunities identified in the prior study. These primitives are not presented as a finalized framework or optimal solution. Instead, they serve as hypotheses of interaction—a way to make otherwise abstract design questions tangible and discussable.

5.1 Mark: Anchoring Reference and Context

The concept of a Mark is designed as a lightweight mechanism for creating referential anchors directly on existing interface elements, addressing users' difficulties in navigating across different

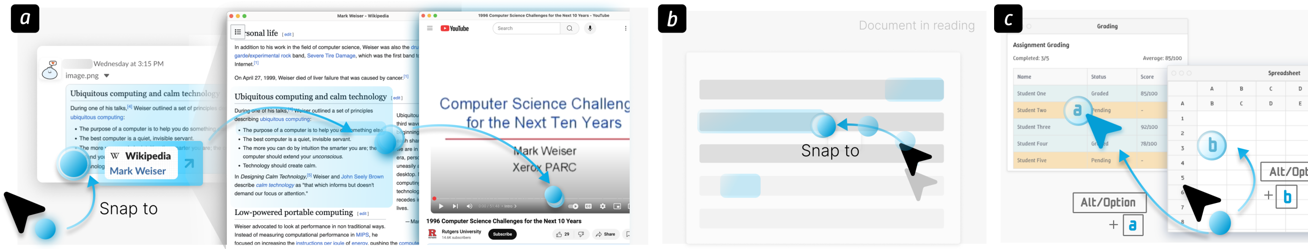


Figure 2: Marks serve as referential anchors to facilitate navigation. a) users can follow a shared mark to the preserved context and navigation history; b) Fairy can latch on nearby objects of interest, allowing quick navigation, such as guiding the users to a desired element on screen; c) marks can facilitate quick back and forth navigation across different view ports via keyboard shortcuts.

contexts within digital environments. A Mark can be created on any user selection, including text spans, interface widgets, windows, hyperlinks, or specific screen locations.

5.1.1 Context and history preservation. Marks preserve the contextual and temporal information associated with a selection. For example, a Mark on a text span records where and when the selection was made, allowing users to save, revisit, or share it with its surrounding context intact (Figure 2a). This supports activities such as online reading, where users often want to reference or share content later without reconstructing its provenance.

5.1.2 Point of interest where fairy latches on. As users move their cursor across the interface, Fairy Cursor can latch onto nearby Marks to indicate relevant content (Figure 2b). This supports faster target acquisition and draws attention to meaningful points in the workspace. For example, when a user moves toward a navigation button, Fairy Cursor can infer intent from the user's ongoing activity and latch onto the button; while reading an online article, Fairy can automatically create Marks on potentially relevant passages to suggest sections worth revisiting. Users can follow these suggestions via shortcuts or ignore them by moving away.

5.1.3 Teleport across time and space. Each Mark is automatically assigned a shortcut that allows users to jump directly to the marked location. This enables rapid navigation across documents and applications without physical cursor movement. In workflows involving back-and-forth navigation—such as grading students' assignments in an educational platform and entering the scores in a spreadsheet—users can quickly switch focus while preserving state at each marked location (Figure 2c).

> Mark probes how a cursor-centric assistant might support reference and orientation without imposing new organizational structures or abstracting content away from its original setting.

5.2 Spell: Reifying Small Actions

The concept of a Spell is designed to explore reifying small, repeated actions as reusable and composable components that are embedded within ongoing interaction flow to address repetitive frictions that do not justify full automation. A Spell represents an action that the Fairy can perform. While some default Spells resemble familiar context-menu actions (e.g., copy, share, search), their key distinction lies in that Fairy may execute the actions proactively based on the context and supporting customization.

5.2.1 Proactive, context-aware actions. Interaction with computing systems can be modeled as a sequence of chunked selection-action (selecting content and then applying an action) or action-selection (choosing a tool and applying it to content) phrases [43, 49]. Fairy Cursor could recognize these granular actions, and compose repetitive sequences as Spells, as well as suggest or execute appropriate Spells. For example, Fairy can memorize the steps a user has taken to collect BibTex citations after repetitive patterns being detected (Figure 3a) and suggest a "Get .bib Entry" spell when a user hovers on a hyperlink across different applications (Figure 3b).

Spells can operate directly on user actions in situations that demand excessive manual input. For example, although drag-and-drop is often considered an intuitive interaction technique, it can become difficult when users must interleave it with application navigation. Fairy can temporarily take control of the dragged item, relieving users from continuously holding down the mouse button and enabling them to use the mouse for complex navigation tasks such as window or tab selection. Similarly, when Fairy Cursor detects complex selection actions—such as dragging to select text across multiple pages—it can temporarily maintain the mouse click as a Spell (Figure 3c). This reduces muscle strain by allowing users to simply move the cursor to indicate the desired selection range.

5.2.2 Customization. Users can define new Spells either by describing the desired behavior in natural language or by demonstration, specifying the situation and the actions to trigger. Once saved, the system can recognize the situation and apply the Spell. For example, the user can invoke the observation mode, and the Fairy Cursor will record users' actions, which can be revised by the user.

> Spell probes how proactive assistance can reduce interaction burden without crossing into unwanted automation, and how users perceive initiative, trust, and reversibility when actions are suggested directly at the cursor.

5.3 Memory: Holding Provisional Information

The concept of Memory aims to support holding ephemeral information without committing to long-term storage. Different from

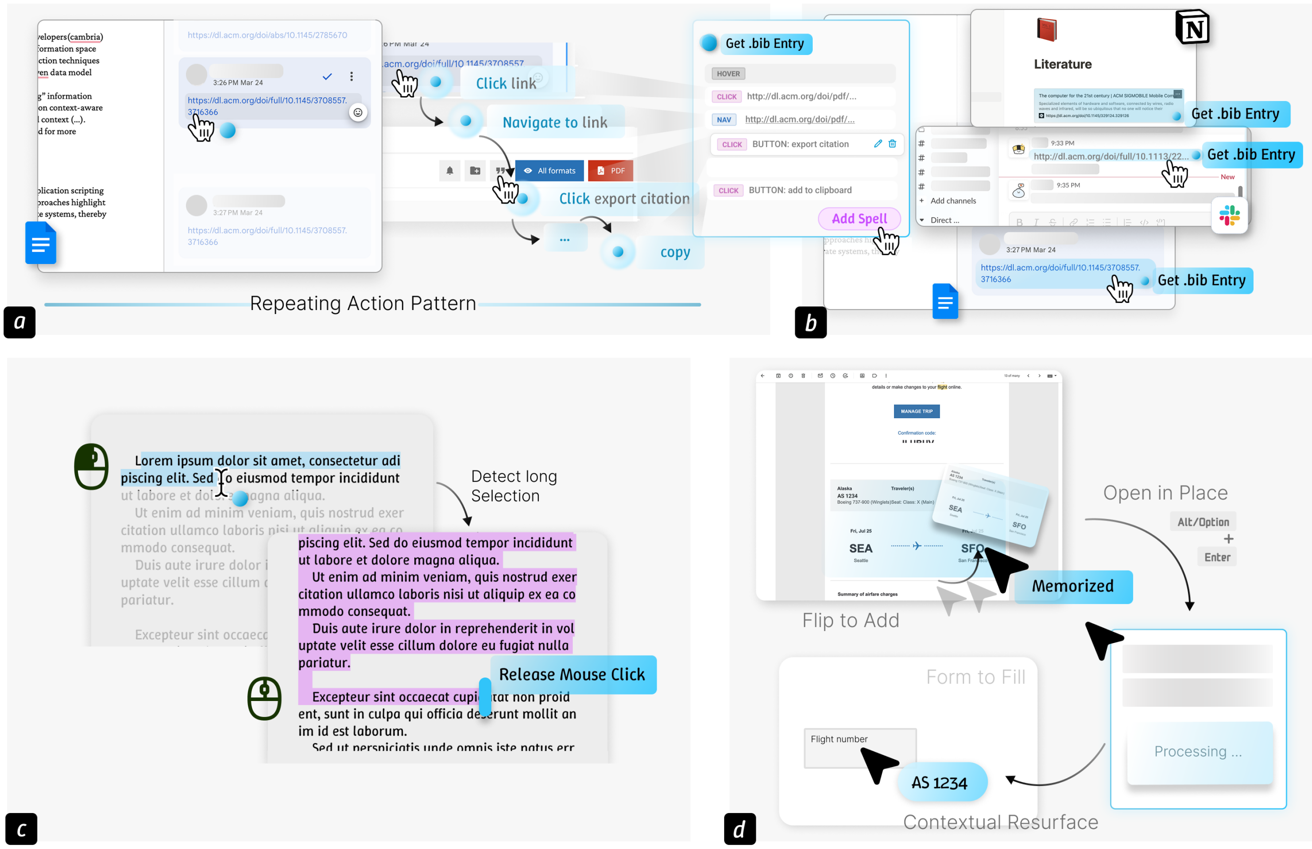


Figure 3: Spell and Memory. Spells are a) recorded, reified, and composable actions, b) which can be suggested by the Fairy. c) They can also be triggered directly when excessive simultaneous inputs are required, such as relieving the users from having to hold the mouse button when selecting across long distance. d) Memory stores ephemeral information and can surface the desired pieces of information in suitable context. Users can use a flip gesture to quickly add a piece of information to memory and Fairy will process the information for intelligent surfacing. Memory is designed as a lightweight, cursor-adjacent floating window, summoned via a keyboard shortcut.

general note-taking tools, Memory is intentionally minimal and cursor-adjacent, supporting momentary holding and opportunistic reuse without requiring decisions about organization or persistence, allowing the users to remain engaged in their ongoing tasks.

5.3.1 Cursor-adjacent working memory. Memory of the Fairy Cursor can appear as a small floating window summoned directly near the cursor via a shortcut. Users can add content to Memory through a simple flip gesture from selections, Marks, or Spell outputs. Unlike most note-taking tools that support complex internal content organization, memory is explicitly designed to be lightweight as a container without complex organizational structures and functionalities. This design signals that Memory is not intended for archival use, but for holding information that is only relevant for the current flow of activity.

5.3.2 Memory as shared context. Because Memory is populated through deliberate user actions reflecting what the users consider relevant, it provides a lightweight way for establishing shared context between the Fairy Cursor and the user. Fairy Cursor leverages this information to modulate its behavior across other primitives—for example, prioritizing where to latch onto Marks, or which Spells to surface in a given situation (Figure 3d).

> Memory explores how a cursor-centric assistant can reduce the "heaviness" of managing information that is only temporarily relevant, without requiring drastic context switching.

5.4 Progressive Actions

To connect these interaction concepts together while balancing noticeability with unobtrusiveness, we adopt progressive disclosure [39] in which keyboard shortcuts reveal increasing levels of interface complexity only as needed. It consists of the following levels as illustrated in Figure 4.

Minimal State: By default, only the Fairy itself is visible, using animations to indicate its status. These include idle (neutral state),

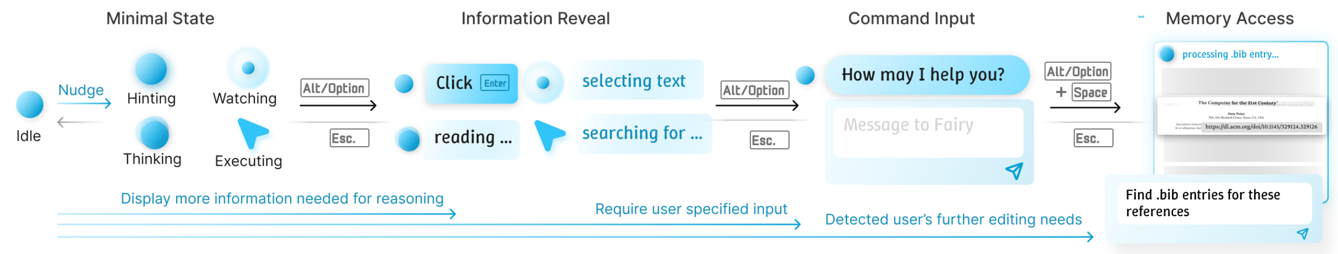


Figure 4: Progressive Action Interaction Model of Fairy Cursor. The user can use keyboard shortcuts to reveal increasing levels of interface complexity as needed.

hinting (indicating available actions), thinking (indicating processing), watching (observing user actions), and executing (showing a blue cursor when manipulating the interface).

Information Reveal: Pressing `Alt/Option` reveals additional information. During thinking, it displays processing progress; during watching, it logs observed user actions; during executing, it documents current operations; and during hinting, it reveals direct action buttons (Spells).

Command Input: If the suggested action is not what the user intended, they can press `Alt/Option` again to bring up a natural language input, allowing users to chat with Fairy Cursor for more customized responses.

Memory Access: `Alt/Option+Space` brings up Fairy's Memory for viewing and editing stored information, with context-specific content displayed in the window.

This progressive disclosure of actions is triggered based on the user's contextual information needs, allowing the Fairy Cursor to provide different levels of support in specific situations. When no assistance is needed, Fairy might appear as just a subtle hinting circle (for instance, briefly pulsing when detecting repetitive text edits to offer formatting assistance). In more complex scenarios, it might directly display the Memory (such as automatically presenting saved receipts when the user opens an expense form). This adaptive behavior ensures that assistance is proportional to task complexity, balancing helpfulness with unobtrusiveness.

5.5 Summary

Together, Mark, Spell, and Memory are interactive primitives centered around the Fairy Cursor, each addressing challenges identified in Study 1. In fact, if only considered in isolation, these primitives are not entirely new to us, they draw on familiar concepts already present in our operating systems, such as bookmarks, macros and clipboard. The uniqueness is that they are now anchored to a single, persistent agent at the cursor. Rather than functioning as discrete tools invoked separately, the three primitives are designed to compose with one another: Marks inform where Spells should be cast and provide the current context for their execution, Memory maintains a broader context that helps resurface relevant Marks and enables reuse of Spells. Moreover, because these primitives are designed for Fairy — an intelligent assistant at the cursor — they are meant to be lightweight, proactive, and context-aware. And since the primitives build on each other, there is a need to manage the resulting complexity, as opposed to relying entirely on user-triggered interactions via a context menu. This motivates the Progressive Actions model, which explores how assistance can be surfaced and negotiated across the three primitives. In the following section, we describe the study we conducted to investigate the interaction experiences with the Fairy Cursor.

6 Study 2: Understanding Cursor-Centric Assistance

The goal of this study is to elicit insights into users' perceptions, challenges, and emerging expectations around a cursor-centric AI assistant. Specifically, we focus on the following research questions:

RQ1 *Presence and Tolerance.* How do users experience the persistent presence of a cursor-following assistant during ongoing work, and what factors influence whether this presence feels helpful, ignorable, or intrusive?

RQ2 *Interpreting Initiative at the Cursor.* How do users interpret initiative from a cursor-centric assistant, and how do they distinguish between helpful suggestions and unwanted interruption?

RQ3 *Making Sense of the Interaction Primitives.* How do users understand and appropriate Mark, Spell, and Memory as forms of in-the-flow assistance, and what expectations or breakdowns emerge in use?

RQ4 *Expectations for Collaboration Dynamics.* What expectations do users form about how a cursor-centric assistant should participate in their work—when it should observe, suggest, or act, and what collaboration roles it could take?

To investigate these questions, we needed a way for participants to directly engage with the proposed designs introduced in Section 5. Ideally, deploying Fairy Cursor at the operating system level would have produced the most authentic findings. However, this was infeasible given the limitations of state-of-the-art AI models and the restricted development support of current operating systems. Existing Vision–Language Model (VLM)–based screen understanding and computer automation techniques are both slow and error-prone: latencies often on the order of tens of seconds, along with errors in perception, inference, or action, would disrupt interaction flow rather than provide assistance. Moreover, contemporary operating systems offer limited support for modifying cursor behavior or enabling fine-grained, cross-application navigation. As a result, our early technical prototypes built atop existing stacks introduced excessive noise, preventing meaningful engagement with the core design concept of Fairy Cursor.

We also considered traditional Wizard-of-Oz techniques, but found them ill-suited to our context. Cursor-centric assistance requires tight temporal coupling between user actions and system responses; human-in-the-loop simulation would be too slow and inconsistent to sustain a believable, in-the-flow interaction. Static mock-ups or video prototypes also cannot capture how users dynamically negotiate attention, initiative, and control over time.

Ultimately, we chose to develop a proof-of-concept environment that allows users to experience plausible and contextually appropriate assistance. We designed four interaction scenarios in which assistance could be initiated without relying on live AI capabilities. Although this approach sacrifices some task authenticity compared to in-the-wild studies, it enables users to reliably control Fairy Cursor and to closely scrutinize the interaction experience, focusing on the four research questions that guide our study.

6.1 The Proof-of-Concept Environment

The environment introduces the interaction primitives incrementally—Mark, Spell, and Memory—reflecting their conceptual dependencies. For each primitive, representative tasks identified from Study 1 were presented, and participants were instructed to complete the tasks using the Fairy Cursor:

Cross-context navigation (Mark). Participants experienced navigation using Marks within and across applications. They were required to leave peripheral notes or anchors on arbitrary regions of a document application (e.g., the margin of a document or a navigation panel), then used those Marks to return to the same context later. They were also asked to navigate between the document application and a spreadsheet application, marking relevant content in one application and using those Marks to relocate, reference, and transform information in another.

Learning repeated actions and guiding automation (Mark + Spell). In an assignment grading scenario, participants repeatedly performed the sequence of leaving feedback, entering a grade, submitting it, and moving to the next student. Fairy Cursor surfaced Spells based on these actions, left automatic Marks on different positions of the interface, allow user to use shortcut to follow the Fairy to the next position and execute the action with guided automation.

Supporting extended selection (Spell). During long text selection tasks, Fairy Cursor detected sustained selection gestures and temporarily held mouse input, allowing participants to continue selecting content by moving the cursor alone.

Collecting information across contexts to fill a form (Memory). Participants collected multiple pieces of information (e.g., names, dates, references) and then reused them to fill out a reimbursement form, experiencing how Memory supports temporary holding and contextual resurfacing.

The environment was deployed as a web application.[1]

6.2 Participants and Study Procedure

We recruited 7 participants for our study through internal communication channels and email outreach with the broader research community. Four participants joined in person, while three participated remotely via Zoom. Each session lasted approximately 70 minutes and was conducted as a facilitated, semi-structured interaction and feedback session. Participants were compensated with a $20 Amazon gift card. The study consists of three phases.

Introduction and Customization (10min). Participants were introduced to the Fairy Cursor concept and invited to customize its visual behavior, including parameters such as following distance and responsiveness to the user's cursor. This initial customization phase helped prime participants' imagination while providing a concrete understanding of the concept.

Hands-on Experience with Fairy Cursor (45min). Participants interacted with Fairy Cursor through a sequence of information work scenarios in the environment. Each interaction concept was demonstrated briefly and then the participants explored it through extended hands-on use. During the process, participants were encouraged to think aloud while exploring, and the facilitator posed targeted questions to probe interpretation, perceived usefulness, moments of friction, and emerging expectations.

Survey and Semi-Structured Interview (15min). Participants completed a detailed survey that assessed their perceptions of Fairy Cursor's presence, communication effectiveness, and practical usefulness. We then conducted semi-structured interviews to gain deeper insights, specifically exploring moments of alignment or tension between user expectations and system behavior, preferences regarding proactive assistance, and participants' visions for future functionality and integration.

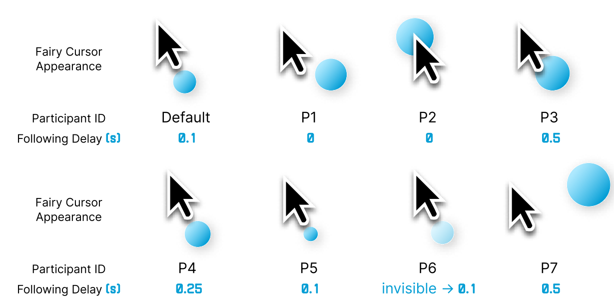


Figure 5: Customized configuration of Fairy Cursor in terms of relative position, size, and following delay.

6.3 Results

6.3.1 Preferred Following Parameters. In the first part of the study, we asked participants to adjust various parameters (relative position, size and following delay) of their Fairy Cursor, as shown in Figure 5. Preferences for the following delay varied: some participants preferred a tightly synced movement (P1, P2), while others favored a more intentional lagged following (P3, P7). Five out of seven participants chose to have the Fairy follow after the cursor rather than before. Notably, P6 initially set the Fairy's size to zero to hide it, but later reverted to the default size after encountering parts of the study that required the Fairy to be visible.

[1] https://fairyplayground.netlify.app/

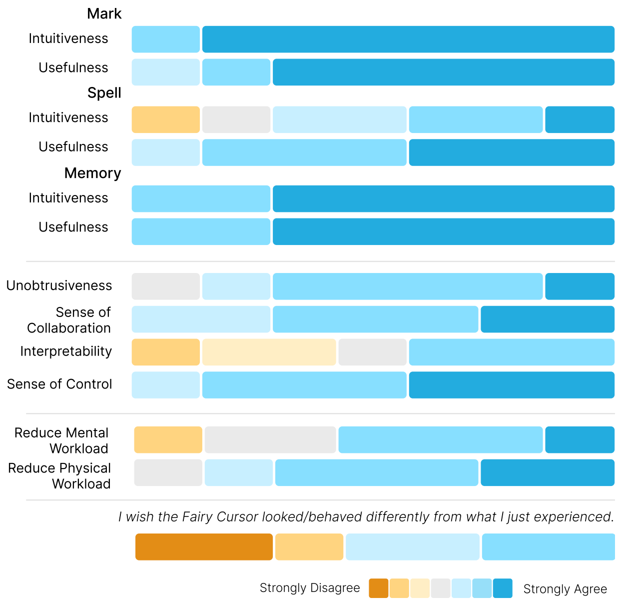


Figure 6: Results of our seven-likert scale questionnaire on participants' perception of Fairy Cursor

6.3.2 Survey Results. We conducted a seven-point Likert scale questionnaire as a complementary instrument to our qualitative findings, rather than as a validation of the system. The survey was designed to elicit participants' self-reported understanding and perceptions of key dimensions of the Fairy Cursor, including the three interaction primitives, the sense of control and collaboration it affords, and their expectations on this new form. A subset of the results is shown in Figure 6 (full questionnaire provided in the supplementary materials). We draw on these responses to contextualize the qualitative findings discussed in the following sections.

6.4 Perception of the Fairy Cursor's Presence [RQ1]

Participants generally reacted positively to the presence of a second following cursor. Most participants did not find this form inherently obtrusive. P6, who initially set the Fairy's size to zero out of concern it would be annoying, later noted, *"In the beginning, I thought it would be annoying, but it turned out not to feel like that."* Several participants reported forgetting about the Fairy until it acted, as P7 mentioned, *"I forgot it was there until it did something."*

After interacting with the Fairy Cursor, participants articulated preferences regarding its movement patterns and proximity to their focus of attention. Many expressed interest in more adaptive, context-sensitive behaviors. For example, P4 suggested that Fairy Cursor should only follow when the primary cursor pauses, while P3 proposed that during typing, the Fairy Cursor should remain nearby, "perched" at the edge of the view port rather than continuously trailing movement.

> *It doesn't have to always chase me. Maybe it could chill in the corner unless I seem stuck.*(P4)

> *It was more helpful when it hovered near where I was working, but not on me.* (P7)

Preferences also diverged around the visibility of this second cursor. Several participants (P3, P4, P6) suggested that the Fairy could be invisible by default and only appear when offering assistance, while others valued its constant presence. P5, in particular, described the persistent following behavior as reassuring:

> *When it's always following, I feel like help is available if I need it.* (P5)

These findings suggest that a cursor-following assistant can coexist with ongoing work without being inherently distracting. However, preference and tolerance of such visual companionship varies across individuals. (Figure 5).

> Effective cursor-centric assistance requires tunable and adaptive presence, rather than a single, fixed design.

6.5 Desire for Clearer Communication of Initiative and System State [RQ2]

The majority of participants reported that the Fairy Cursor's communication was generally understandable and appropriately subtle. They also appreciated the sense of controllability afforded by the progressive actions, noting that it allowed them to engage with assistance at their own pace (P5). However, participants also reported moments of hesitation or confusion when the Fairy acted without making its intent or consequences sufficiently clear. These breakdowns clustered around two unresolved forms of ambiguity:

6.5.1 Ambiguity around internal state. One recurring concern was the lack of visibility into what the Fairy was doing "behind the scenes." Participants occasionally wondered whether it was remembering things, preparing suggestions, or simply idle.

> *I wasn't sure if it was remembering what I wrote or just hovering there.* (P2)

This lack of transparency made it difficult for users to decide whether to invoke Fairy for assistance. Participants suggested lightweight feedback mechanisms, such as transient animations or subtle visual cues, to indicate background activity without introducing pop-ups or explicit notifications.

> *If it's doing something for me (when added to working memory), I want a little sign—definitely not a pop-up, but something that lets me know it worked.* (P4)

6.5.2 Ambiguity around feedforward and consequences. Confusion also arose where users lacked clear feedforward about what would happen if they followed the Fairy's lead. For example, when the Fairy hovered over a button in the grading scenario, participants were uncertain whether following it would trigger a click action or merely change the position. This uncertainty often made users reluctant to act.

> *If it's leading me somewhere, I want to know what happens if I follow-like will this button be clicked by Fairy Cursor?* (P6)

These findings highlight a central tension for cursor-centric assistance: effective communication of initiative is both critical and difficult. Because the assistant operates near the cursor, where visual space is limited and distractions are costly, insufficient signaling can undermine even well-timed and correct assistance. Participants' responses suggest that interpreting initiative depends on multiple dimensions of communication, including its intent, processing state and likely outcomes.

Designing a richer yet unobtrusive visual language for expressing state, progress, and consequences is essential for cursor-centric assistants to be perceived as helpful.

6.6 Usefulness and Limitations of Interaction Primitives [RQ3]

Participants generally found the core interaction primitives—Mark, Spell, and Memory—to be expressive and useful. When asked to imagine additional scenarios where they might use these primitives, participants articulated key aspects they expected each primitive to support, as reported below.

6.6.1 Mark. Mark was described as a useful and lightweight way to hold or highlight information references. Participants related it to "*digital Post-Its*" (P3). Its value in contextual navigation was frequently mentioned as useful (P1, P2, P3, P6, P7). At the same time, participants expressed a desire for Marks to behave in a more content-aware manner. For instance, P7 imagined they would use the Mark when watching a video lesson and taking notes, but noted that when returning to a previously marked location in the notes, they expected Marks to move with the associated text rather than remain fixed at their original positions.

> *I want the Mark to follow the content where my cursor left off, not where I placed it originally.* (P7)

This suggests that in information work, users sometimes need positional references and at other times need content-based references. To be effective, Marks need to support both purposes by adapting to what is being marked and maintaining references as content evolves.

6.6.2 Spell. Spell was the least immediately intuitive concept. Participants found Spells that are surfaced as suggested actions were easier to understand, whereas those that were proactively embedded in the interaction flow were less intuitive.

A recurring theme was the importance of timing. Because Spells could be placed within rapid, low-level action sequences, participants were sensitive not only to delay, but also to being too early. For example, in the grading scenario, the Fairy sometimes jumped ahead to the next action (e.g., moving toward the grading button) while participants were still composing feedback. Although the suggestion was correct, several participants found the premature movement disruptive, as it shifted attention away from their current focus (P1, P2). Conversely, some of the participants' actions were performed so quickly that even accurate suggestions arrived too late to be useful. In the extended text selection scenario, P3 noted that they often completed the selection action before noticing the Fairy's suggestion of holding the mouse click. These cases highlight that for supporting low-level actions, there is often a narrow temporal window in which proactive assistance is helpful without causing confusion.

Participants also reflected on how proactive assistance should scale over time. P5 described that they enjoyed step-by-step guidance initially but expected more consolidated assistance after repeated interactions:

> *I like following it at first, but after a few rounds, I expect it to just give me a complete version that I can accept and move on.* (P5)

This suggests that participants expect Spells to evolve over time. Initial, guided assistance helped establish trust and understanding, but over time, users expected encapsulating repeated steps into higher-level actions.

6.6.3 Memory. The concept of Memory was perceived to be the easiest to understand. P3 and P4 described it as a powerful, extended clipboard. All participants appreciated the idea of storing and resurfacing content across contexts. Several noted that the system's explicit constraints helped them form accurate expectations about what the Fairy could and could not do.

Because participants understood Fairy Cursor as an AI-enhanced system, they expected Memory to exhibit some degree of intelligent organization. P1, P3, and P4 anticipated the Memory to display stored information differently when invoked in a form-filling scenario versus during writing. Others suggested adding lightweight organizational functionality, such as tagging or pinning memory items to control recall and reuse (P2, P6), or search functionality for fast retrieval (P1, P7).

Cursor-centric assistance benefits from proactiveness and contextual awareness, and is highly sensitive to temporal precision and interaction cost. Progressive adaptation and multimodal control are critical for sustaining usefulness over time.

6.7 Expectations for Collaboration Dynamics [RQ4]

Overall, participants found Fairy Cursor created a sense of collaboration rather than simple tool use (3 agree, 2 strongly agree), aligning with prior characterizations of systems experienced as collaborative counterparts rather than passive instruments [2, 26, 36]. Rather than expecting independent action, participants emphasized turn-taking, suggesting that the Fairy should time its participation based on users' ongoing actions and readiness. P6 proposed that the Fairy leverage action-informed turn-taking models instead of acting immediately when an opportunity is detected.

Building on earlier findings, participants emphasized that effective collaboration depends on sensitivity to task context and user state. They expected the Fairy's level of initiative to vary dynamically with cognitive load and task demands. For cognitively intensive work, such as focused writing or reasoning, participants preferred the Fairy to remain nearby and supportive rather than proactive (P3, P7). In contrast, for routine or repetitive tasks, they welcomed more active assistance.

> *It would be cool if Fairy was more active when I'm doing boring tasks but backed off during focused writing.* (P2)

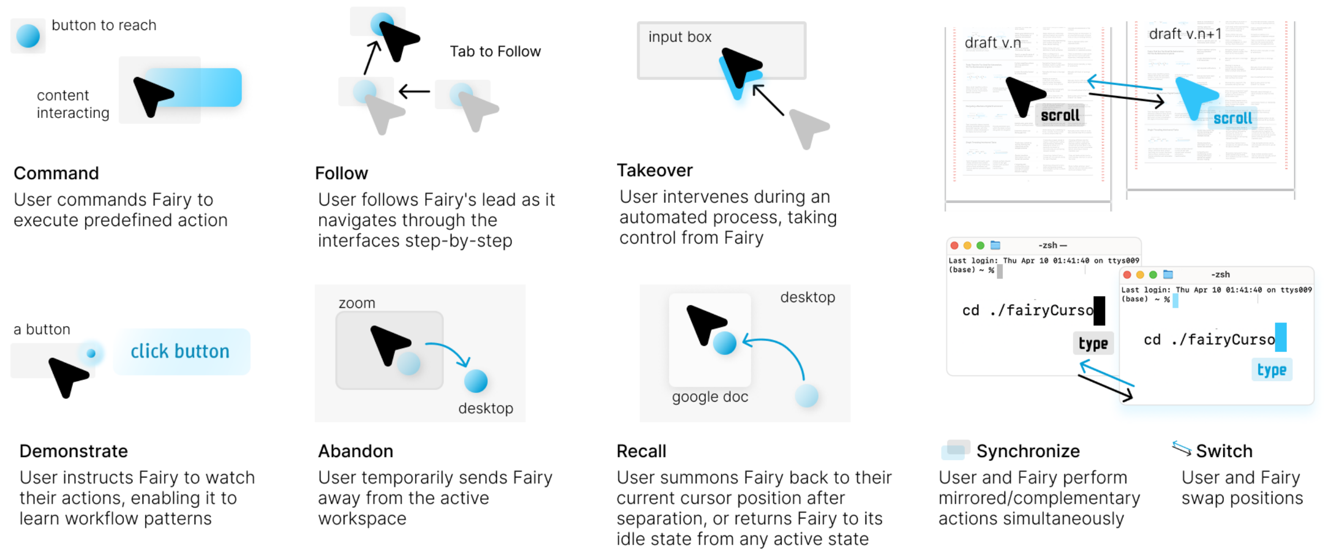


Figure 7: Eight collaboration dynamics between Fairy Cursor and the user's cursor, synthesized from participants' reported desired forms of collaboration with the Fairy Cursor.

Personalization emerged as a central expectation. Because Fairy Cursor maintains a persistent, always-present form factor, participants implicitly treated it as a personal agent that should evolve over time. Multiple participants expressed interest in a version of Fairy Cursor that adapts to their individual habits, recurring breakdowns, and work rhythms.

> *I don't want it jumping in all the time, but if it learned that I always forget to rename files after saving, that would be useful.* (P1)

> *If it knows when I go to see the score of this baseball game, I always navigate this way, then it can guide me and I can just tab through.* (P3)

We summarized eight collaboration dynamics based on how participants envisioned coordinating control and initiative with a cursor-centric assistant (Figure 7), reflecting their desires to flexibly shift between observing, being guided, acting jointly, and working independently, depending on tasks and personal preferences.

> Effective cursor-centric assistance requires a repertoire of collaboration dynamics that adapt to task context, cognitive load, and individual habits. Because the assistant is always present, users expect it to learn over time rather than act uniformly.

6.8 Limitations: What We Failed to Probe

As an exploratory study conducted in a simulated task environment, our evaluation was designed to surface early user reactions, expectations, and breakdowns around cursor-centric assistance. While this approach allowed us to probe interpretability and interaction dynamics in a controlled setting, it also limited our ability to assess several critical aspects of long-term, real-world use. Below, we outline key limitations that emerged from the study and motivate directions for future investigation.

6.8.1 Triggering Appropriate Levels of Proactivity. A key open question is when and how the Fairy should take initiative. While participants could articulate preferences for more or less proactive behavior in specific moments, it remains unclear how to systematically detect these boundaries. We were unable to derive generalizable principles from the short-term, simulated setting. Future work based on real-world usage data and longer engagement can result in more accurate initiative thresholds.

6.8.2 Uncertainty About Real Unobtrusiveness. Because the study was conducted in a simulated environment with mock tasks, we were not able to assess whether the Fairy Cursor truly reduces cognitive or manual load in meaningful, sustained ways. In such an environment, assistance is "expected to happen"; therefore, multiple participants mentioned that it is hard to evaluate whether it would be the same case under real settings:

> *It does not give wrong suggestions now, but if it does, I'm not sure how I'd feel about it.* (P2)
> *... Actually, I said I wanted this and that, but I'm not sure—maybe in real life I'd just want it to do the simplest thing.* (P7)
> *Now you are interacting with it for almost an hour, but in real case, you probably only interacting with it 2 percent of your time... so i guess it's fine [in terms of obtrusiveness].* (P3)

While participants imagined several benefits of the Fairy Cursor, we could not verify these outcomes with the current setup. A deployment study will be needed to validate the assistant's utility across different real-world workflows.

6.8.3 Interaction Modality for Controlling the Second Cursor. Because effective interaction with the Fairy Cursor requires coordinating between the primary cursor and a secondary assistive agent, some participants found it effortful. Additionally, the current design relies heavily on keyboard shortcuts to trigger actions. While this supported efficiency for some users, many participants (P1, P2, P6, P7) found the shortcuts difficult to remember or unintuitive, which discouraged deeper engagement with the system. Participants therefore expressed interest in alternative input modalities, including gesture-based interactions (P3, P4, P7) or lightweight voice input (P5, P7).

6.8.4 Interaction Noise and Visual Clutter. While most participants tolerated—or even appreciated—the Fairy Cursor's visual presence, a few raised concerns about potential visual noise and long-term fatigue. This highlights a broader issue around interaction traces: overlapping animations, unintended triggering, and subtle cognitive interference could become problematic in more complex or fast-paced real-world scenarios. P4 reflected on a past experience with a cursor customization tool:

> *There was a tool that let you decorate your cursor with eyes and stuff... it didn't do anything functional, so it wasn't annoying. At first it was cool, and I used it for a while—but eventually I just turned it off. I don't know why—I just wanted my original cursor back.* (P4)

This comparison suggests that even when the assistant is visually appealing and non-disruptive, subtle friction or fatigue may accumulate over time, leading users to revert to minimal setups. These concerns did not manifest strongly during our short-term study and point to the needs for longitudinal studies in more realistic, sustained workflows.

6.8.5 Can Users Form a Habit? Our study could not probe how deeply ingrained interaction habits affect or can be affected by the adoption of cursor-centric assistance. Prior to developing the proof-of-concept environment, we deployed a prototype built with Electron that operated directly on macOS, scoped to a copy-and-paste workflow: detecting a copy action, predicting the paste destination, and guiding the user there with potential follow-up actions. During piloting, however, we encountered something similar to active user paradox [11] for micro-tasks: users had already begun navigating to their paste destination the moment they copied—no one waited for the Fairy to suggest where to go and what to do next. The actions we aimed to assist were so habituated that assistance arrived 'too late' even with minimal latency.

Yet interaction habits do shift when new capabilities prove reliably valuable. Code autocompletion, for example, was unnecessary when developers typed fluently on their own—but once tools like Copilot [20] demonstrated consistent value, many developers adapted their workflows to pause and wait for suggestions. A similar dynamic may apply to cursor-centric assistance: if the Fairy can reliably anticipate the right action, users may learn to leave room for it. Whether such habituation occurs, and whether it genuinely reduces effort rather than merely redistributing it, are questions that can only be answered through reliable AI intelligence and longitudinal deployment studies.

7 Implications for Future Work

In this section, we reflect on the broader implications of our findings and outline future research directions on cursor-centric assistance.

7.1 Evolving and Personalized Fairy Cursor with Rich Visual Language

The cursor we use every day has taken decades to evolve into its current form, along with its various behaviors. The simple blue dot trailing behind the mouse cursor in our Fairy Cursor prototype represents just an initial exploration in this new direction, and we expect it to continue evolving over time. Findings from our study suggest that this form factor has substantial room to evolve, both visually and behaviorally. Unlike the traditional cursor, which functions primarily as a passive pointing indicator, the Fairy Cursor embodies an active assistant—one that continuously observes user behavior and adapts to ongoing activities and preferences. This opens an opportunity for an ongoing channel for grounding [16], allowing the user and the assistant to incrementally build shared understanding of context, intent, and progress through the course of interaction. This connects with a growing body of HCI research on how to establish and maintain common ground between humans and AI systems [9, 37, 45, 46]. The cursor-following form factor offers a promising site for such grounding, however, as our study suggests, how to reify such communication into effective interaction remains an open challenge.

Future work should explore how an expanded visual and interactive language can better support this paradigm. This includes richer visual forms, context-sensitive behaviors, and more expressive communication mechanisms such as subtle animations, micro-gestures, or ambient feedback that convey intent without demanding attention. Equally important is deeper personalization: allowing users to tailor the Fairy Cursor's appearance, interaction patterns, and degrees of autonomy to match their preferences and tolerance for intervention. Long-term deployment studies will be critical for understanding how users' relationships with such an agent evolve over time, what levels of intervention remain acceptable, and how to design for trust, transparency, and sustained user control.

7.2 Integration with Operating System and AI

The idealized version of Fairy Cursor is a personal assistant operating at the OS level, equipped with context understanding capabilities. This requires an in-depth exploration of how to integrate AI into the operating system, along with the broader implications of doing so. For instance, detecting recurring patterns and recommending suitable actions would require the Fairy Cursor to have knowledge of the interface elements the user interacts with. One approach is to have all applications provide information to an AI interpretation module to facilitate context understanding; however, this would require application developers' compliance, which may be challenging to enforce. Alternatively, the Fairy Cursor could record users' screens and infer action patterns based on their interaction history. However, even if recordings are stored and processed locally, having an AI agent that continuously sees, records, and analyzes everything displayed on a user's screen could still raise significant privacy concerns.

More broadly, cursor-centric assistance challenges interaction conventions that have been in place since the Xerox Alto: what the cursor is, how initiative is distributed, how assistance is invoked [35]. Studying departures from such deeply rooted paradigms requires methods that go beyond one-time evaluations: longitudinal deployments that track how interaction patterns, user habits, and trust evolve over sustained real-world use. Future work will need to explore hybrid approaches that balance feasibility, effectiveness, and privacy. For example, lightweight metadata sharing combined with selective screen access could enable sufficient context understanding without complete visual surveillance. Another important area is the development of transparent interaction models, where users are kept aware of when and how their behaviors are being observed and can easily adjust the Fairy Cursor's visibility, autonomy, and data collection settings. For example, a simple 'abandoned' action would send the Fairy away and stop its access to content, or the users could instruct the Fairy Cursor to aggressively avoid privacy intrusion.

7.3 The Adoption of Proactive AI Agents

Despite decades of research and development, the successful adoption of proactive AI companions in real-world settings remains elusive. Microsoft's Clippy, for example, has become a cautionary tale: it was overly intrusive, insensitive to context, and ultimately perceived as a distraction and a failure [17]. Much of this failure stemmed from the limited ability of AI at the time to robustly understand user context.

Although contemporary AI models are substantially more capable, their performance on open-ended, real-world computer tasks remains far from reliable. For example, OpenAI's Computer Use Agent reports a success rate of approximately 38.1% on representative benchmarks of general computer interaction, with higher performance on narrower, web-based tasks but persistent failures in complex, unconstrained settings [41]. In proactive scenarios, such error rates pose a significant risk: incorrect or mistimed interventions can quickly erode user trust and recreate the frustrations. This risk is particularly acute for cursor-centric assistance. As our findings show, even small misaligned responses can interrupt users' flow and turn potentially helpful assistance into a distraction.

These limitations suggest that progress in proactive assistance may depend less on pursuing general autonomy and more on carefully scoped, task-specific support. Rather than attempting to anticipate everything a user needs, a promising strategy is to design for assistance that is intentionally limited in scope—ensuring that each assistive action provides clear, local value. In this sense, successful proactive agents may be defined not by how much they can do, but by how consistently they provide assistance that is just enough.

7.4 Integrating with Future Information Workspaces

Our ultimate goal is to enable users to stay in the flow while engaging with information tasks. This research rests on a core assumption: users are motivated to engage directly with their tasks, even as AI systems grow more powerful. Our findings support this assumption. In the retrospective think-aloud study, participants frequently described being immersed in their work despite the inefficiencies. This observation suggests that preserving direct engagement with one's information, rather than fully delegating to AI, may be an important consideration for future assistant design.

Our efforts therefore focus on stripping away unnecessary overhead while making each interaction within the workspace more rewarding. Through the study, many of the inefficiencies we identified are shaped or exacerbated by the prevailing document-and-application-centered interface paradigm that is well known for creating fragmentation [34]. In today's systems, information is increasingly encapsulated in silos, forcing users to manually shuttle files between applications, cobble together workflows, and endure frequent context switching. The growing reliance on chat-based interfaces further compounds this fragmentation by introducing yet another isolated workspace. These disruptions increase attentional demands, elevate cognitive load, and complicate collaboration. Recent work suggests that the rise of AI models offers a unique opportunity to transform this outdated paradigm into a generative, malleable information space, particularly when AI is embodied in the form of a personal assistant [10]. Such a shift in the underlying "interface physics" would fundamentally alter the action space and contextual understanding capability of the cursor-following assistance. We are excited about future directions that integrate generative and malleable information workspaces with the Fairy Cursor to explore how AI can be fully leveraged to reshape how human information activities are supported and experienced.

8 Conclusion

We envision a type of digital assistants that provide lightweight, interleaved support embedded directly within users' ongoing activities, rather than isolated in separate windows. This work explores one such form factor—a cursor-following agent—and investigates its potential to provide in-the-flow assistance. Through a multi-stage design inquiry, we examine both where such assistance is needed and how it might be designed. Our retrospective think-aloud study reveals that disruptions to flow often stem from small, cumulative frictions embedded in interleaved work. These include referring to prior context, repeating low-level actions, and managing ephemeral information. We instantiated these observed needs in Fairy Cursor as a design probe. User study with a proof-of-concept environment surfaces both opportunities and tensions: participants valued micro-task support that helped them maintain momentum, while also articulating expectations around appropriate timing, contextual awareness, visual expressiveness, and personalization. We see cursor-centric assistants as a promising direction for human–AI collaboration that complements existing forms of assistance. Future systems that build on this work can explore task-scoped contextual understanding, expressive yet subtle visual language and diverse collaboration dynamics. More broadly, this work points toward personal agents that are not destinations we go to for help, but companions that stay with us—supporting micro-tasks, ephemeral information, and interleaved workflows as they unfold.

References

[1] [n. d.]. DIA Browser. https://www.diabrowser.com/. Accessed: 2025-04-08.
[2] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. 2019. Guidelines for Human-AI Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in*

Computing Systems (Glasgow, Scotland UK) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3290605.3300233
[3] Anthropic. 2024. Computer Use. https://docs.anthropic.com/en/docs/agents-and-tools/computer-use. Accessed: 2026-03-01.
[4] Apple Inc. 1987. Knowledge Navigator. https://en.wikipedia.org/wiki/Knowledge_Navigator Concept video and vision for future technology.
[5] Brian P Bailey and Joseph A Konstan. 2006. On the need for attention-aware systems: Measuring effects of interruption on task performance, error rate, and affective state. *Computers in human behavior* 22, 4 (2006), 685–708. doi:10.1016/j.chb.2005.12.009
[6] Ravin Balakrishnan, Thomas Baudel, Gordon Kurtenbach, and George Fitzmaurice. 1997. The Rockin'Mouse: integral 3D manipulation on a plane. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '97)*. Association for Computing Machinery, New York, NY, USA, 311–318. doi:10.1145/258549.258778
[7] Frederik Brudy, David Ledo, Michel Pahud, Nathalie Henry Riche, Christian Holz, Anand Waghmare, Hemant Bhaskar Surale, Marcus Peinado, Xiaokuan Zhang, Shannon Joyner, Badrish Chandramouli, Umar Farooq Minhas, Jonathan Goldstein, William Buxton, and Ken Hinckley. 2020. SurfaceFleet: Exploring Distributed Interactions Unbounded from Device, Application, User, and Time. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 7–21. doi:10.1145/3379337.3415874
[8] Yining Cao, Jane L E, Chen Zhu-Tian, and Haijun Xia. 2023. DataParticles: Block-based and Language-oriented Authoring of Animated Unit Visualizations. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 808, 15 pages. doi:10.1145/3544548.3581472
[9] Yining Cao, Yiyi Huang, Anh Truong, Hijung Valentina Shin, and Haijun Xia. 2025. Compositional Structures as Substrates for Human-AI Co-creation Environment: A Design Approach and A Case Study. In *CHI Conference on Human Factors in Computing Systems (CHI '25)* (Yokohama, Japan). ACM, New York, NY, USA, 25. doi:10.1145/3706598.3713401
[10] Yining Cao, Peiling Jiang, and Haijun Xia. 2025. Generative and Malleable User Interfaces with Generative and Evolving Task-Driven Data Model. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 686, 20 pages. doi:10.1145/3706598.3713285
[11] John M Carroll and Mary Beth Rosson. 1987. Paradox of the active user. Interfacing thought: cognitive aspects of human-computer interaction. *JM Carroll. Cambridge, MA, the MIT Press: pp* (1987), 80–111.
[12] Joseph Chee Chang, Nathan Hahn, Yongsung Kim, Julina Coupland, Bradley Breneisen, Hannah S Kim, John Hwong, and Aniket Kittur. 2021. When the Tab Comes Due:Challenges in the Cost Structure of Browser Tab Usage. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 148, 15 pages. doi:10.1145/3411764.3445585
[13] Joseph Chee Chang, Yongsung Kim, Victor Miller, Michael Xieyang Liu, Brad A Myers, and Aniket Kittur. 2021. Tabs.do: Task-Centric Browser Tab Management. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '21)*. Association for Computing Machinery, New York, NY, USA, 663–676. doi:10.1145/3472749.3474777
[14] Kerry Shih-Ping Chang and Brad A. Myers. 2016. Using and Exploring Hierarchical Data in Spreadsheets. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 2497–2507. doi:10.1145/2858036.2858430
[15] Mon Chu Chen, John R. Anderson, and Myeong Ho Sohn. 2001. What can a mouse cursor tell us more? correlation of eye/mouse movements on web browsing. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems* (Seattle, Washington) *(CHI EA '01)*. Association for Computing Machinery, New York, NY, USA, 281–282. doi:10.1145/634067.634234
[16] Herbert H Clark and Susan E Brennan. 1991. Grounding in communication. American Psychological Association, 127–149. doi:10.1037/10096-006
[17] Microsoft Corporation. 1997. Clippy (Clippit). Virtual assistant in Microsoft Office. Discontinued in 2007.
[18] Douglas C. Engelbart. 1968. The Mother of All Demos. Computer demonstration at the Fall Joint Computer Conference. Available at https://www.dougengelbart.org/content/view/209/448/.
[19] Figma Inc. 2016. Figma. https://www.figma.com/. Accessed: 2025-04-08.
[20] GitHub. 2021. GitHub Copilot. https://github.com/features/copilot. Accessed: 2026-03-01.
[21] Victor M. González and Gloria Mark. 2004. "Constant, constant, multi-tasking craziness": managing multiple working spheres. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vienna, Austria) *(CHI '04)*. Association for Computing Machinery, New York, NY, USA, 113–120. doi:10.1145/985692.985707
[22] Google LLC. 2006. Google Docs. https://www.google.com/docs/about/. Accessed: 2025-04-08.
[23] Grammarly, Inc. 2023. Grammarly: AI Writing Assistance. https://www.grammarly.com. Accessed: 2026-01-20.
[24] Tovi Grossman and Ravin Balakrishnan. 2005. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Portland, Oregon, USA) *(CHI '05)*. Association for Computing Machinery, New York, NY, USA, 281–290. doi:10.1145/1054972.1055012
[25] Tovi Grossman, Ken Hinckley, Patrick Baudisch, Maneesh Agrawala, and Ravin Balakrishnan. 2006. Hover widgets: using the tracking state to extend the capabilities of pen-operated devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montréal, Québec, Canada) *(CHI '06)*. Association for Computing Machinery, New York, NY, USA, 861–870. doi:10.1145/1124772.1124898
[26] Marc Hassenzahl, Jan Borchers, Susanne Boll, Astrid Rosenthal-von der Pütten, and Volker Wulf. 2020. Otherware: how to best interact with autonomous systems. *Interactions* 28, 1 (Dec. 2020), 54–57. doi:10.1145/3436942
[27] Jeffrey Heer. 2019. Agency plus automation: Designing artificial intelligence into interactive systems. *Proceedings of the National Academy of Sciences* 116, 6 (2019), 1844–1850. doi:10.1073/pnas.1807184115
[28] Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Pittsburgh, Pennsylvania, USA) *(CHI '99)*. Association for Computing Machinery, New York, NY, USA, 159–166. doi:10.1145/302979.303030
[29] Jeff Huang, Ryen White, and Georg Buscher. 2012. User see, user point: gaze and cursor alignment in web search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) *(CHI '12)*. Association for Computing Machinery, New York, NY, USA, 1341–1350. doi:10.1145/2207676.2208591
[30] Edwin Hutchins. 1995. *Cognition in the Wild*. MIT press.
[31] Edwin L. Hutchins, James D. Hollan, and Donald A. Norman. 1985. Direct Manipulation Interfaces. *Hum. Comput. Interact.* 1, 4 (1985), 311–338. doi:10.1207/s15327051hci0104_2
[32] Apple Inc. 2011. Siri. https://www.apple.com/siri/. Accessed: 2026-03-01.
[33] Sean Kandel, Andreas Paepcke, Joseph Hellerstein, and Jeffrey Heer. 2011. Wrangler: interactive visual specification of data transformation scripts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. Association for Computing Machinery, New York, NY, USA, 3363–3372. doi:10.1145/1978942.1979444
[34] Victor Kaptelinin and Bonnie A Nardi. 2009. *Acting with technology: Activity theory and interaction design*. MIT press.
[35] Ralph Kimball and B Verplank E Harslem. 1982. Designing the Star user interface. *Byte* 7, 1982 (1982), 242–282.
[36] Joseph CR Licklider. 1960. Man-computer symbiosis. *IRE transactions on human factors in electronics* 1 (1960), 4–11.
[37] Xingyu Bruce Liu, Shitao Fang, Weiyan Shi, Chien-Sheng Wu, Takeo Igarashi, and Xiang 'Anthony' Chen. 2025. Proactive Conversational Agents with Inner Thoughts. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 184, 19 pages. doi:10.1145/3706598.3713760
[38] Gloria Mark, Victor M. Gonzalez, and Justin Harris. 2005. No task left behind? examining the nature of fragmented work. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Portland, Oregon, USA) *(CHI '05)*. Association for Computing Machinery, New York, NY, USA, 321–330. doi:10.1145/1054972.1055017
[39] Jakob Nielsen. 2006. *Progressive Disclosure*. Nielsen Norman Group. https://www.nngroup.com/articles/progressive-disclosure/ Accessed: 2026-01-18.
[40] OpenAI. 2022. ChatGPT. https://openai.com/chatgpt. Accessed: 2026-03-01.
[41] OpenAI. 2025. Computer-Using Agent. https://openai.com/index/computer-using-agent/. Accessed: 2025-04-09.
[42] OpenAI. 2025. Operator. https://openai.com/index/introducing-operator/. Accessed: 2026-03-01.
[43] Jef Raskin. 2000. *The humane interface: new directions for designing interactive systems*. Addison-Wesley Professional.
[44] Ben Shneiderman and Pattie Maes. 1997. Direct manipulation vs. interface agents. *Interactions* 4, 6 (Nov. 1997), 42–61. doi:10.1145/267505.267514
[45] Priyan Vaithilingam, Ian Arawjo, and Elena L. Glassman. 2024. Imagining a Future of Designing with AI: Dynamic Grounding, Constructive Negotiation, and Sustainable Motivation. In *Proceedings of the 2024 ACM Designing Interactive Systems Conference* (Copenhagen, Denmark) *(DIS '24)*. Association for Computing Machinery, New York, NY, USA, 289–300. doi:10.1145/3643834.3661525
[46] Priyan Vaithilingam, Munyeong Kim, Frida-Cecilia Acosta-Parenteau, Daniel Lee, Amine Mhedhbi, Elena L. Glassman, and Ian Arawjo. 2025. Semantic Commit: Helping Users Update Intent Specifications for AI Memory at Scale. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25)*. Association for Computing Machinery, New York, NY, USA, Article 137, 18 pages. doi:10.1145/3746059.3747778

[47] Stephen Voida, Elizabeth D. Mynatt, and W. Keith Edwards. 2008. Re-framing the desktop interface around the activities of knowledge work. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology* (Monterey, CA, USA) *(UIST '08)*. Association for Computing Machinery, New York, NY, USA, 211–220. doi:10.1145/1449715.1449751

[48] Steve Whittaker. 2011. Personal information management: from information consumption to curation. *Annual review of information science and technology* 45, 1 (2011), 1. doi:10.1002/aris.2011.1440450108

[49] Haijun Xia, Ken Hinckley, Michel Pahud, Xiao Tu, and Bill Buxton. 2017. WritLarge: Ink Unleashed by Unified Scope, Action, & Zoom *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 3227–3240. doi:10.1145/3025453.3025664

[50] Haijun Xia, Tony Wang, Aditya Gunturu, Peiling Jiang, William Duan, and Xiaoshuo Yao. 2023. CrossTalk: Intelligent Substrates for Language-Oriented Interaction in Video-Based Communication and Collaboration. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA, USA) *(UIST '23)*. Association for Computing Machinery, New York, NY, USA, Article 60, 16 pages. doi:10.1145/3586183.3606773

[51] Chun Yu, Yuanchun Shi, Ravin Balakrishnan, Xiangliang Meng, Yue Suo, Mingming Fan, and Yongqiang Qin. 2010. The satellite cursor: achieving MAGIC pointing without gaze tracking using multiple cursors. In *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA) *(UIST '10)*. Association for Computing Machinery, New York, NY, USA, 163–172. doi:10.1145/1866029.1866056

[52] Amy X. Zhang, Michael Muller, and Dakuo Wang. 2020. How do Data Science Workers Collaborate? Roles, Workflows, and Tools. *Proc. ACM Hum.-Comput. Interact.* 4, CSCW1, Article 22 (may 2020), 23 pages. doi:10.1145/3392826

[53] Chen Zhu-Tian and Haijun Xia. 2022. CrossData: Leveraging Text-Data Connections for Authoring Data Documents. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 95, 15 pages. doi:10.1145/3491102.3517485