PointAloud: An Interaction Suite for AI-Supported Pointer-Centric Think-Aloud Computing

Frederic Gmeiner*
Autodesk Research
Toronto, ON, Canada
Carnegie Mellon University
Pittsburgh, PA, USA
gmeiner@cmu.edu

John Thompson
Autodesk Research
Atlanta, GA, USA
john.thompson@autodesk.com

George Fitzmaurice
Autodesk Research
Toronto, ON, Canada
george.fitzmaurice@autodesk.com

Justin Matejka
Autodesk Research
Toronto, ON, Canada
justin.matejka@autodesk.com

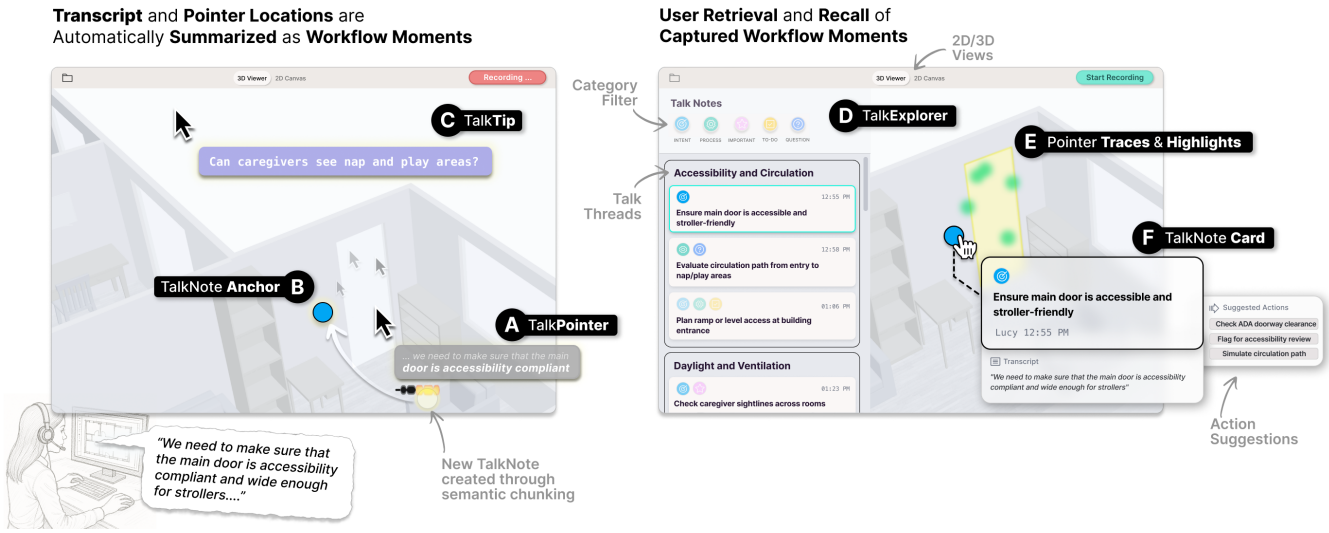


Figure 1: PointAloud allows users to (1) automatically capture their think-aloud verbalizations and pointer locations while working on architectural software tasks in 2D and 3D; with (A) TalkPointer providing pointer-adjacent low-distraction real-time feedback on the capture process and indicating when a new TalkNote gets created, the TalkNote is (B) contextually anchored in the design canvas; Additionally, (C) TalkTips provide short proactive system suggestions in response to users' activities. (2) To retrieve captured moments, (D) TalkExplorer provides a topically-clustered list view with filter options; when selecting TalkNotes, (E) captured pointer traces and relevant design elements are highlighted in the canvas, along with the TalkNote's (F) card, which features transcript, summary, process labels, and system-suggested follow-up actions.

Abstract

Think-Aloud Computing, a method for capturing users' verbalized thoughts during software tasks, allows eliciting rich contextual insights into evolving intentions, struggles, and decision-making processes of users in real-time. However, existing approaches face practical challenges: users often lack awareness of what is captured by the system, are not effectively encouraged to speak, and miss or are interrupted by system feedback. Additionally, thinking aloud should feel worthwhile for users due to the gained contextual AI assistance. To better support and harness Think-Aloud Computing, we introduce PointAloud, a suite of novel AI-driven pointer-centric interactions for in-the-moment verbalization encouragement, low-distraction system feedback, and contextually rich work process documentation alongside proactive AI assistance. Our user study

*Work done as an intern researcher at Autodesk Research.

CHI '26, Barcelona, Spain

ACM ISBN 979-8-4007-2278-3/26/04
https://doi.org/10.1145/3772318.3790797

with 12 participants provides insights into the value of pointer-centric think-aloud computing for work process documentation and human-AI co-creation. We conclude by discussing the broader implications of our findings and design considerations for pointer-centric and AI-supported Think-Aloud Computing workflows.

CCS Concepts

• **Human-centered computing → Interaction paradigms**.

Keywords

think-aloud computing, work process documentation, human-AI interaction, pointer interactions, context-aware support

ACM Reference Format:

Frederic Gmeiner, John Thompson, George Fitzmaurice, and Justin Matejka. 2026. PointAloud: An Interaction Suite for AI-Supported Pointer-Centric Think-Aloud Computing. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26), April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 37 pages. https://doi.org/10.1145/3772318.3790797

1 Introduction

While digital design tools support the *production* of creative work, they offer little support for documenting the user's *processes* that generate them. In domains such as architectural planning or engineering design, critical decisions often unfold tacitly through iterative cycles of exploration and reflection-in-action, shaped by evolving ideas, constraints, and partial understandings [36, 59, 66]. Despite their centrality to professional practice, these situated decision-making processes are difficult to capture, revisit, or communicate [24, 28]. However, externalizing and documenting such fleeting thoughts and rationales is valuable: it can support individual reflection [10, 65, 66], preserve design rationales for later retrieval [24, 28], facilitate communication with collaborators [13, 26], and even provide rich contextual data for more adaptive AI-driven design support [19].

Building on this need, prior work has started to unpack **Think-Aloud Computing** [38] as a technique to capture knowledge in situ by encouraging users to verbalize their reasoning while working. However, previous approaches and interfaces for think-aloud computing face important challenges:

(1) Users often lack awareness of what is being captured;
(2) Users may not be effectively encouraged to verbalize their thoughts in the moment;
(3) Users' thought verbalization does not receive real-time responses from the system, or such system feedback can feel either too disruptive or too subtle to notice; and
(4) The additional effort from users to verbalize their thoughts should lead to worthwhile design assistance from the system.

In this paper, we introduce **PointAloud**, a suite of novel AI-driven pointer-centric interactions that address these challenges by embedding think-aloud support directly into the visual and interactive fabric of design tools. These interaction techniques support capturing, representing, and responding to users' verbalizations in real-time as they work on a design task.

A core concept of the suite is **`TalkPointer`**: Augmenting the space around the user's mouse pointer with a dynamic display to provide real-time feedback and system responses without causing workflow distractions. This way, users can receive low-distraction in-the-moment feedback from the system in response to their verbalization and screen actions.

Another core part of PointAloud is **`TalkNotes`**: real-time, semantically parsed annotations that automatically transform users' spoken thoughts into lightweight "sticky notes" anchored directly to relevant areas of the workspace. When a user later selects a `TalkNote`, the system highlights the note together with the documented pointer activity and related design elements from the moment of verbalization, resituating the user's reasoning in its original context.

In this way, PointAloud bridges the gap between ephemeral verbalization and persistent design knowledge, supporting both in-the-moment sensemaking and longer-term documentation of process. At the same time, by externalizing and capturing users' evolving intentions, concerns, and rationales, such interactions open **new opportunities for more context-aware forms of human–AI co-creation**, where AI support incorporates awareness of how the user's thinking and reasoning develops over time, instead of responding only to momentary cues.

To further probe the potential of these interaction techniques, we developed the **PointAloud System**: a CAD application for inspecting and annotating architectural floor plans in both 2D and 3D. The system allows us to test the PointAloud interaction suite within the concrete design application of architectural planning. Under the hood, the system leverages real-time transcription and a large language model (LLM) to segment, semantically cluster, summarize, categorize, and visually contextualize verbalized thoughts to create `TalkNotes` that can be reviewed later. The system extends beyond capture by providing tools for grouping related `TalkNotes` into thematic clusters, resurfacing notes when similar contexts arise in the design process, and offering dynamic actions for users to follow up within the workspace.

To explore the benefits and limitations of PointAloud, we conducted a **user study** with 12 professionals. The study comprised various architectural planning-related tasks that participants completed using the PointAloud prototype system. In comparative tasks, the participants worked with a reduced feature set of the prototype as a baseline, mimicking a conventional text-based live transcription interface. The baseline interface, without components embedded around the cursor or canvas, offers a low-distraction, minimal think-aloud condition.

Our findings indicate that users perceived stronger support for making use of their spoken thoughts and for keeping track of their design process with PointAloud compared to the baseline. Participants especially appreciated how PointAloud helped externalize fleeting ideas into automatically structured and contextualized `TalkNotes`, resurface earlier reasoning during later recap, and receive low-distraction, workflow-embedded AI suggestions.

In summation, this paper makes three main contributions:

(1) **PointAloud:** a suite of pointer-centric interaction techniques for AI-supported think-aloud computing, design workflow documentation, and context-aware human-AI co-creation;

(2) **User study insights** on how PointAloud interactions support concurrent thought verbalization, work process documentation, and human-AI co-creation, by using a PointAloud-based CAD prototype system as a technical probe;
(3) **Design considerations** for future pointer-centric and AI-supported think-aloud computing interactions, such as strategies for incentivizing verbalization, designing pointer-ambient displays, enabling more process-aware human–AI co-creation, and embedding documentation seamlessly within users' ongoing workflows.

Our work focuses on pointer-centric think-aloud interactions for supporting the creative activities of architectural designers, which we examine in depth throughout this paper. At the same time, the design patterns embodied in PointAloud offer an applicable foundation that researchers and practitioners can adapt across diverse software workflows. In doing so, PointAloud introduces new interaction techniques for documenting work processes and enabling richer forms of human-AI co-creation.

2 Related Work

2.1 Pointer-Centric Display Techniques

In traditional "point-and-click" WIMP-style interfaces, **pointer-adjacent tool tips** or **right-click context menus** are often used to surface options relevant to the hovered/selected/dragged object or GUI element [1, 52]. Beyond such static pop-ups, research has also proposed dynamic and multimodal tool tips that adapt to user state or task context to provide pointer-adjacent multimedia assistance for learning feature-rich software applications [15, 22, 48]. In recent years, many commercial software applications, such as Adobe Photoshop or Autodesk Fusion, have also integrated animated tool tips that demonstrate feature usage to improve their learnability.

From a cognitive perspective, these forms of pointer-proximate augmentation **reduce mental load** by minimizing the cost of attention switching while supporting parallel information processing [2, 50, 55, 71]. In WIMP-style and pointer-centric interfaces, a common assumption is that **users' visual attention is close to their pointer**. Eye–pointer alignment studies provide empirical support for this assumption, and even suggest that the cursor can serve as an attention proxy with accuracy comparable to gaze tracking [8, 32, 40].

While a large body of work has explored improving the usability or accuracy of pointers as *input* techniques (for example to improve accessibility [73], gestures [11], editing details [58] or 3D pointing [76]), little work has further explored augmenting the space around the pointer as a low-distraction *output* technique.

Prior work in telepresence research has suggested conveying the presence, intention, and identity of collaborators in real-time, remote software workflows through **displaying remote users' mouse cursors** [25]. In recent years, the emergence of commercial collaborative shared canvas tools such as Miro or Figma has allowed users to see each other's cursors on the same canvas when collaborating in real-time, including chat bubbles pinned to collaborators' cursors [14, 51]. Similarly, recent research systems like *"Pointer Assistant"* [61] propose LLM-driven pointers to represent agents in GUI workflows to guide users' focus and enhance human-AI co-creation.

Our work builds upon these strands of work of pointer-centric display techniques by contributing a technique to "stick" dynamic visual feedback and suggestions adjacent to the pointer based on the user's captured verbalization and pointer movements during their design workflows. Through this, we aim to **(1) provide a low-distraction display** of real-time think-aloud verbalization feedback and proactive system responses while also **(2) capturing users' pointer locations during workflows** for enabling integrated process documentation and context-rich user models for improved human-AI co-creation.

2.2 Context-Aware Support Systems

A second strand of research has explored how systems can automatically surface timely and relevant guidance, suggestions, or feedback by modeling the user's task context and goals. Such **context-aware support systems** have been proposed to help users navigate complex, feature-rich software environments that often overwhelm novices [35, 46]. For example, *DiscoverySpace* suggests task-level action macros in Photoshop, allowing novices to maintain confidence and discover features more effectively [16]. In addition to surfacing commands, several systems have explored how to provide contextualized tutorials and examples. *TutoriVR* extends 2D video tutorials with embedded 3D contextual aids to support VR painting [68]. *RePlay* gathers activity across applications to suggest video tutorials, helping users reduce time spent on web search [17]. Similarly, *Shöwn* dynamically presents conceptual examples during comic drawing [54]. Recent systems also integrate lightweight AI-generated commentary, such as *FeedQUAC*, which provides persona-driven ambient feedback to support reflection and inspiration in creative workflows [42].

Another related body of work highlights the role of **ambient or peripheral displays** [60] for surfacing contextual knowledge and resources in the background of the primary user activity. For instance, *Ambient Help* presents automatic, context-sensitive videos and help resources on a secondary display to support opportunistic learning without harming productivity [49]. Similarly, *SidePoint* integrates a peripheral knowledge panel to surface concise, relevant knowledge items alongside the slide presentation authoring activities [41]. Another example is *InterWeave*, which embeds contextual search suggestions directly into users' note-taking environments, helping participants connect new information to their emergent structures of knowledge [56]. Together, these works show how presenting ambient contextual support, such as side panels and peripheral displays, can reduce interruption costs and embed guidance into users' ongoing, primary workflows.

Finally, research on **mixed-initiative systems** has articulated how proactive system feedback can be balanced with the user's notion of control. Early principles emphasize the coupling of automated services with direct manipulation in ways that preserve user agency; exemplified by Bayesian user models for inferring users' goals and needs from observed actions and queries [30, 31]. Subsequent domain-specific systems have extended these ideas, such as *Soloist*, which leverages deep-learning audio processing to generate customizable tutorials and provide real-time feedback for guitar learning [70].

Across these streams, context-aware support systems demonstrate different strategies for helping users: inferring goals and surfacing relevant functionality, providing tutorials and examples in situ, embedding knowledge in peripheral displays, and balancing initiative between the system and the user. Building atop this prior work, we contribute novel interaction techniques for proactive, AI-driven assistance grounded in multimodal user context models elicited through capturing users' concurrent think-aloud verbalization and pointer activities.

2.3 Workflow Capture, Documentation, and Retrieval Systems

A long-standing line of work in HCI has examined how digital systems can capture, document, and later retrieve knowledge about work processes, design decisions, and task histories. For example, research has explored systems that **capture and document workflows** to create rich histories toward understanding and reuse. *Chronicle* [23] records the complete creation history of graphical documents by linking its content to the steps, tools, and settings used, thereby turning the resulting replayable document into a rich archive and learning resource. Similarly, *Co-notate* [64] explores real-time annotations that combine audio, video, and textual traces to capture evolving situational knowledge during collaborative design activities. More recently, *Meta-Manager* allows automatically collecting and organizing provenance information in software development by tracking code changes, including AI-generated or copy-pasted fragments, enabling developers to later answer questions about unfamiliar code bases and design rationale [29]. Other approaches augment physical fabrication activities by capturing multimodal sensor data about users, tools, and expertise to enable personalized feedback and contextual analysis of making processes [20].

Another strand has investigated **documenting work processes through demonstration** and automated tutorial generation. For example, *Grabler et al.* [21] introduced a system that automatically generates step-by-step tutorials from demonstrations in photo manipulation software. *Torta* [53] extends this approach by combining operating-system-wide activity tracing with screencasts to generate mixed-media tutorials. Crowd-powered systems such as *StreamWiki* [43] leverage audiences of live knowledge-sharing streams to collaboratively generate archival documentation in real time.

Systems have also explored **capturing the contextual "why"** (rationale) behind work artifacts [24, 28]. For example, *Callisto* links conversational chat traces with computational notebook elements to surface the rationale behind data analytic workflows [69], while other systems capture and link online discussion threads to the code under debate [57]. These approaches highlight the importance of recording not only what was done but also *the reasoning behind design decisions*.

Complementary work has addressed the challenge of **retrieving captured process information** for reflection and sensemaking through novel interfaces. For example, *Delta* allows visualizing and comparing alternative software workflows at multiple levels of granularity to help users understand trade-offs [37]. Similarly, *Tesseract* introduces spatial querying of design recordings through a Worlds-in-Miniature interface, enabling expressive search of past multimodal design activities [45]. In the context of online meetings, *TalkTraces* [7] provides real-time capture and visualization of relevant discussed content, while *MeetingVis* [67] uses visual narratives to support remembering meeting content and context.

Other systems have explored **implicit multimodal process documentation** by capturing users' speech and gaze activities. For example, *GAVIN* combines gaze with voice input to anchor voice notes to text passages [34], while follow-up work explores gaze and speech more broadly for implicit multimodal interaction design [33].

Similarly, recent research by *Krosnick et al.* has proposed the concept of **Think-Aloud Computing** [38]: using concurrent verbalization as a lightweight but rich mechanism for documenting users' work processes. The authors extend the traditional think-aloud protocol into everyday computing, prompting users to speak while working and contextualizing this speech with system state. Their findings show that think-aloud computing captures subtle design intent, rationale, and process knowledge that traditional documentation often misses, while requiring comparable effort. By situating knowledge capture in natural verbalization rather than post hoc reporting, think-aloud computing represents a promising direction for low-effort and rich workflow documentation.

Building atop this rich body of work on capturing, documenting, and retrieving workflows, our contribution lies in extending the vision of Think-Aloud Computing by contributing AI-supported interaction techniques that are tightly integrated into pointer-centric, think-aloud design workflows, as detailed in the next section.

3 Exploring Interaction Design Principles for Supporting Think-Aloud Computing

Our work is grounded in the concept of *Think-Aloud Computing* [38]—a low-effort method for systems to elicit users' knowledge in situ by encouraging them to verbalize their reasoning while engaging in software tasks. Such verbalizations can surface rich, contextual insights into users' evolving intentions, struggles, and decision-making processes in real time, which are otherwise difficult to capture by only observing their interactions with graphical user interfaces.

Especially in the context of design-related activities, such as in architectural planning or engineering design, critical decisions often unfold tacitly through cycles of exploration and reflection-in-action, shaped by evolving ideas, constraints, and partial understandings [36, 59, 66]. These situated processes are notoriously difficult to document for later revisitation or communication [24, 28]. Think-aloud computing, however, has the potential to externalize and record such fleeting rationales and can thus support self-reflection [10, 65, 66], preserve design justifications for future retrieval [24, 28], and facilitate collaboration [13, 26].

Beyond documentation, capturing verbalized thoughts **also creates opportunities for more effective human–AI co-creation** [18, 19]. Users' spoken reasoning can provide context-rich data and serve as a foundation for AI systems to deliver more aligned and situated assistance. At the same time, prior approaches to think-aloud computing interfaces face several challenges: (1) users often lack awareness of what is actually being captured, (2) they may not feel

sufficiently encouraged to articulate their thoughts in the moment, (3) system responses to verbalizations are frequently either absent, overly disruptive, or too subtle to notice, and (4) the additional effort of speaking aloud must ultimately translate into tangible and worthwhile assistance for the user.

These challenges raised several guiding questions for our design exploration:

(1) *How might we **design systems** that **actively incentivize and sustain thinking aloud**?*
(2) *What complementary **contextual data**, alongside verbalizations, should be captured to **represent user processes more holistically**?*
(3) *How can **in-the-moment feedback** be delivered **without interrupting** or distracting users' workflows?*
(4) *In what ways can **captured process data** be harnessed to **foster** more effective **human–AI co-creation**?*

To address these questions, we engaged in a set of formative activities: (1) a literature review on workflow capture, documentation, and retrieval systems, as well as on context-aware support and low-distraction feedback techniques *(see Section 2)*; and (2) a four-week ideation phase involving iterative cycles of sketching and refining interaction concepts, supported by daily check-ins of the first and second authors and weekly meetings with the broader research team, where feedback was also gathered from a larger group of HCI researchers. Across this period, we generated over 15 distinct design concepts and iteratively distilled them into a smaller set of directions, all documented on a shared Miro board. Through this process, we distilled our explorations into four overarching *design principles*, detailed below:

3.1 Design Principles

Based on our guiding questions and formative activities, we developed the following interaction design principles for AI-supported think-aloud computing in the context of design-related tasks:

DP1 Subtle yet perceivable feedback on user context capture. To increase users' awareness and trust in the capture process, the system should provide real-time visual feedback on both verbalization activity and live transcription. Such feedback must be lightweight: Noticeable enough to reassure users about what is being captured, but subtle enough to avoid interrupting the primary design workflow. This ensures that users feel confident their thinking is being recorded without suffering unnecessary distraction.

DP2 Automatically document the user's design process as structured and contextually rich moments. While the user works and verbalizes, the system should automatically detect and document distinct moments of the design process. Each documented moment should combine the user's verbalization with contextual process information *(such as pointer activity, screenshots, and linked design elements)* so that captured moments form a richer record of the workflow. In doing so, the system creates a durable memory of the process that supports later continuity, reflection, and recall, while also providing a data foundation (user models) for more context-aligned AI-suggestions, enabling more effective human–AI co-creation.

DP3 Provide diverse mechanisms to represent and retrieve documented moments. To support sensemaking and provenance, the system should support users in flexibly revisiting and working with previously documented moments. Captured moments should be represented and accessible in multiple complementary forms — such as in-context overlays in the 2D/3D design workspace or thematically clustered in side-panel lists. Retrieval should be further supported through categorization and filtering mechanisms, allowing users to navigate thematic aspects of their workflow.

DP4 Offer proactive, context-aware prompts and follow-up actions to the user. To encourage verbalization and foster human-AI co-creation, the system should proactively respond to users' speech with context-aware prompts and suggested actions. For example, when silence occurs, it may nudge users to elaborate on relevant topics or dynamically resurface past notes that connect to the current situation. The system should also suggest situation-specific follow-up actions, enabling designers to fluidly extend and act upon prior lines of thought as part of their design process.

4 PointAloud System: A Pointer-Centric AI-supported Think-Aloud Computing System

Informed by our formative activities and the design principles for supporting Think-Aloud Computing described in the previous section, we designed a *suite* of novel pointer-centric interaction techniques and adaptive user interface (UI) elements we coined *PointAloud*.

To probe the potential of this collection of techniques in a concrete design-related task context, we developed the **PointAloud System**: a GenAI-driven CAD application for inspecting and annotating architectural floor plans in 2D and reviewing corresponding 3D models. The prototype embeds the PointAloud interaction suite directly into the design workflow, allowing us to explore how pointer-centric think-aloud computing can support process documentation, reflection, and AI-assisted co-creation (Figure 2).

In this section, we demonstrate the utility of *PointAloud* by first illustrating its functionalities within brief use-case scenarios. We then follow with a detailed description of the proposed interface mechanisms and their implementation.

4.1 Example Use Case Scenarios

Lucy is an architect tasked with converting an existing apartment into a licensed childcare micro-center for toddlers. She opens the **PointAloud** system to begin exploring the client-provided floor plan.

4.1.1 *2D Use Case: Annotating the floor plan.*

Lucy loads the provided 2D floor plan and 3D model of the apartment into the workspace. To capture her reasoning as she works, she presses the *"Start Transcription"* button.

(1a) Talking Through Early Design Concerns: As she pans across the floor plan, she verbalizes her initial thoughts: *"We need*

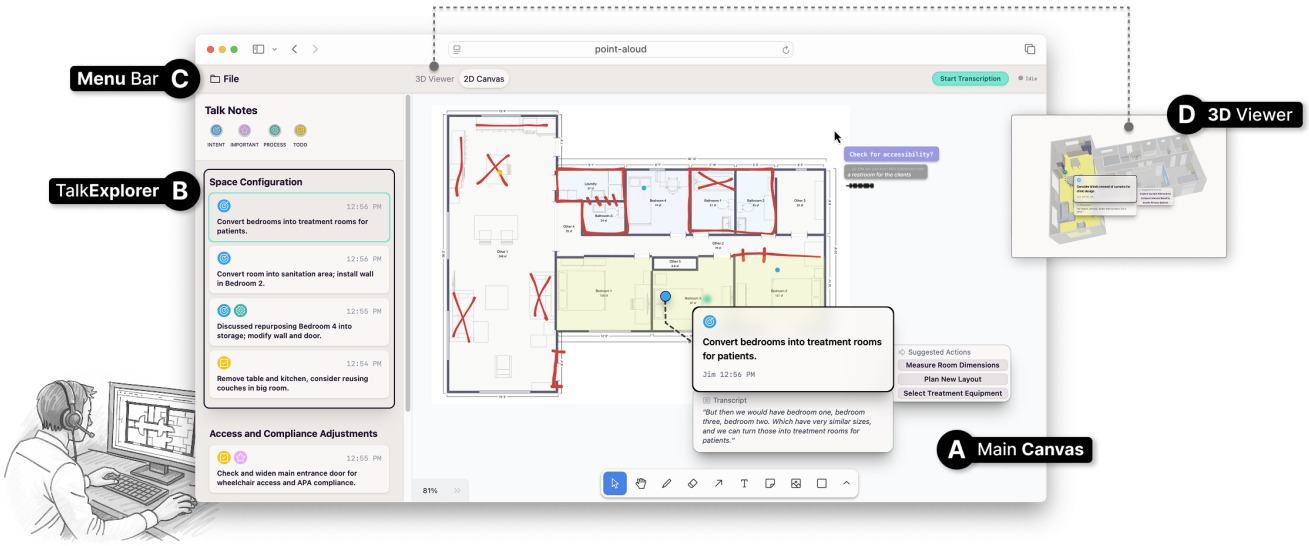


Figure 2: Screenshot of the *PointAloud* system: (A) Main canvas with activated 2D view containing sketches on a floor plan with an unfolded TalkNote card and TalkPointer next to the mouse cursor; (B) TalkExplorer sidebar for browsing and filtering of captured TalkNotes; (C) Menu bar with controls for starting/stopping transcription and switching between 2D and (D) 3D view.

to make sure that the main door is accessibility compliant and wide enough for strollers." Her speech appears live as a small text element next to her mouse pointer **(TalkText)**. A soft, bubble-style indicator grows as she continues talking, letting her see that the system is actively listening without pulling her out of the design task **(TalkVis)**.

(1b) Automatic TalkNote Creation: When Lucy shifts focus to another concern—*"These two interior walls might be load-bearing; I'll need to check structural drawings before removing them."*—the system recognizes this as a new line of thought and generates a **TalkNote**. A small colored dot appears on the floor plan at the entrance she was pointing to, directly anchoring the TalkNote within the spatial design context. Each anchor's color corresponds to the TalkNote's process category label (e.g., light blue for *Design Intent*, magenta for *Important*, dark blue for *Questions*), giving Lucy a quick visual cue of the note's type at a glance.

(2) Revisiting and Exploring TalkNotes: Hovering over the dot expands it into a **TalkNote Card**, containing the transcript snippet, an automatically generated summary ("Ensure main door is accessible and stroller-friendly"), and a category label (*Design Intent*). Alongside the card, Lucy sees a faint yellow highlight over the entrance door and traces of her pointer movements during that utterance, situating the note in its original design context. Lucy hovers over another previously created TalkNote about natural light and sees her earlier pointer traces near the living room windows. This helps her quickly recall why she considered converting the space into the main play area to maximize daylight.

(3) Receiving Proactive System Suggestions (TalkTips): As she continues sketching and talking, a subtle prompt appears near her pointer: "`Can caregivers see nap and play areas?`" This is a system-generated **TalkTip**. Lucy responds aloud: *"That's an important point—I'll need to make sure caregivers can supervise both areas."* The system records her response and creates a new TalkNote tagged as *Important* and *Design Intent*, anchoring it near the playroom partition wall.

(4) Surfacing Related Prior Notes (TalkReminders): While reflecting on possible layouts for the nap room, two earlier TalkNotes reappear on the floorplan, highlighted in red and briefly summarized next to their anchors. The system has detected that Lucy's current verbalization relates to these earlier concerns, resurfacing them as **TalkReminders**. Lucy states her appreciation for the prompt since it jogs her memory about a prior decision on daylight and supervision that is again relevant to the new arrangement.

(5) Triggering Contextual Follow-Up Actions (TalkNote Action Suggestions): When Lucy revisits the accessibility note at the entrance, it contains a contextual action suggestion: "`Check ADA doorway clearance.`" She clicks the small button, which automatically overlays the required clearance radius onto the floor plan. This helps her visually confirm that the current doorway is too narrow and flags the element with a subtle red outline. For another TalkNote about daylight, the action menu offers to "`Simulate daylight exposure.`" Lucy accepts, and the workspace briefly shades the floor plan according to window orientation, helping her evaluate which areas will receive the most natural light. These contextual follow-up actions extend TalkNotes from passive memory cues into active design supports, giving Lucy lightweight but targeted tools right when they are relevant. After continuing for 20 minutes, Lucy stops the transcription and saves the project, which now contains 55 TalkNotes.

4.1.2 ***3D Use Case: Preparing for a client meeting.*** The next day, Lucy re-opens the project in **PointAloud** to prepare for an upcoming meeting with her clients. In the **TalkExplorer** side panel, she sees all of her previously captured TalkNotes automatically clustered by theme, such as *Accessibility & Circulation*, *Daylight & Ventilation*, and *Supervision & Safety*. When hovering over a note in the side panel, its corresponding anchor reappears in the 2D/3D workspace, showing the highlighted design elements and her pointer traces from the moment of verbalization. This allows Lucy to quickly re-situate her earlier reasoning in the design context.

To organize her discussion points, Lucy filters the TalkExplorer to show only notes labeled as *Design Intent*, *Important*, *Questions*, and *To-Dos*. She then uses these filtered notes to draft her meeting document, capturing early design decisions, unresolved questions, and key concerns such as supervision sightlines and acoustic separation in the nap room. This way, her spontaneous thoughts from the exploration session are transformed into a structured, filterable work process documentation to draw upon as she crafts a focused agenda for client discussion.

4.2 Interface Features

In this subsection, we describe the main interaction features of the *PointAloud Interaction Suite*. Together, these features demonstrate how pointer-centric feedback and persistent contextual representations can support designers in capturing, revisiting, and extending their thinking while working on design tasks. We developed the *PointAloud System* to instantiate these interactions and interfaces with architectural planning as a proxy design task. The prototype provides two spatial planning tools: a 2D floor plan tool and 3D model inspector tool. Both tools provide interaction mechanisms to support pointer-centric think-aloud computing.

4.2.1 ***System Interface Overview.*** The interface of the *PointAloud System* (Figure 2) consists of three main components:

(1) **Main Canvas:** The central workspace for sketching in a 2D floor plan view and navigating corresponding 3D models.
(2) **TalkExplorer Sidebar:** A panel for browsing, clustering, and filtering captured TalkNotes, supporting organization and review of verbalized ideas.
(3) **Menu bar:** Controls for starting or stopping transcription, switching between 2D and 3D views, and accessing file operations.

Figure 3: Pointer-adjacent TalkPointer display comprising TalkTip, TalkText, and TalkViz

4.2.2 ***TalkPointer: Dynamic Feedback Anchored to the Cursor (DP1, DP4).*** To provide lightweight but continuous feedback on the think-aloud process, the system features the **TalkPointer**: a dynamic display anchored right of the user's mouse cursor (Figure 3). By visually linking proactive system messages and feedback about the user's speech activity to their current pointer location, TalkPointer provides process-related cues and reassures users that their verbalizations are being captured and contextualized without requiring them to divert attention from the design task.
TalkPointer integrates three complementary components:

Figure 4

(1) TalkText (DP1): A short transcription overlay that streams the user's most recently captured words in real time, providing immediate feedback on the system's ongoing speech transcription.

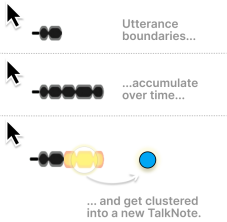


Figure 5

(2) TalkViz (DP1): Visual indicators that signal utterance boundaries and chunking operations, allowing users to see how/when their speech has been segmented and clustered into new TalkNotes.

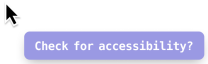


Figure 6

(3) TalkTip (DP1, DP4): Brief, context-sensitive prompts that appear both during pauses and in response to users' speech. They encourage continued verbalization, surface relevant considerations, or pose open-ended questions to support deeper reflection.

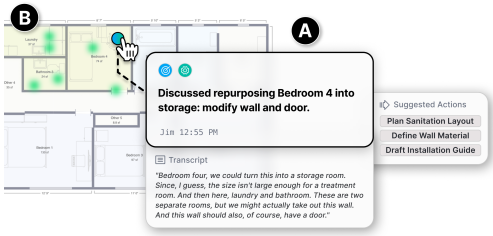


Figure 7: An unfolded TalkNote with two key components: (A) structured note content combining the user's transcript, system-generated summary, process labels, and suggested follow-up actions; and (B) spatial anchoring that situates the note at the location where the user was pointing during verbalization, complemented by pointer traces (green dots) and design element highlights (yellow overlays).

4.2.3 **TalkNotes: Contextualized Representations of Design Moments (DP2, DP3, DP4).** PointAloud captures segments of speech as **TalkNotes**: persistent, structured units that combine transcribed text with contextual metadata. TalkNotes are generated automatically when the system detects a shift in topic or intent, allowing designers to externalize their reasoning without manual effort. Each TalkNote contains the original transcript, a concise generated summary, and process category labels (Figure 7).

Figure 8

Process Label Categories (DP2): Each TalkNote is automatically assigned one or more categories, reflecting different kinds of design reasoning: *Design Intent* (high-level goals or rationales), *Process* (operations, tools, or workflow steps), *ToDo* (tasks to complete later), *Important* (flagged critical information), *Problem* (issues or obstacles), and *Question* (open uncertainties).

Beyond textual content, TalkNotes are anchored to the design workspace through multiple forms of contextualization:

Figure 9

Spatial Anchors (DP3): Notes appear as overlays on the canvas at the location where the user was pointing during verbalization (as 2D/3D location).

Figure 10

Pointer Traces (DP2, DP3): Visual paths of cursor movement during speech are stored and shown as overlays, providing contextual grounding.

Figure 11

Design Element Highlights (DP2, DP3): Relevant architectural elements referenced during users' speech are visually linked to the note.

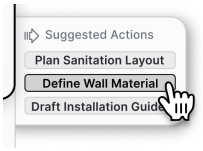


Figure 12

Action Suggestions (DP4): Based on the TalkNote's captured context, the system dynamically generates a UI button menu with follow-up system actions for users to trigger[1].

To make these representations lightweight and accessible, each TalkNote appears initially as a small anchor on the canvas. Hovering over or selecting the anchor expands it into a **TalkNote Card**, showing the transcript snippet, summary, and category labels (Figure 7 A). This provides designers with a quick way to revisit and reflect on earlier reasoning in its original design context.

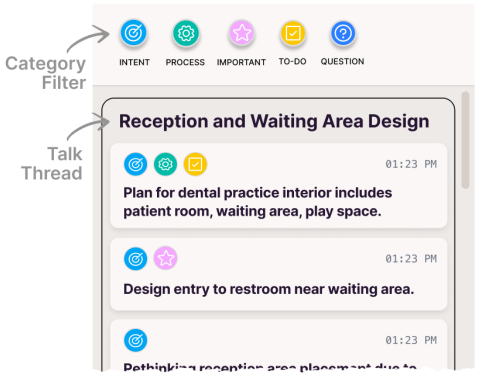


Figure 13: TalkExplorer sidebar with filter options and TalkNotes topically clustered into TalkThreads.

4.2.4 **TalkExplorer: Browsing, Clustering, and Filtering Notes** *(DP3).* For more deliberate review and organization, PointAloud provides the **TalkExplorer**: a sidebar list view for navigating captured TalkNotes (Figure 13). Notes are dynamically organized into **TalkThreads**, which cluster related ideas based on semantic similarity and temporal proximity. Users can filter displayed TalkNotes to focus on specific categories (e.g., *Design Intent*, *Question*, *To-Do*).

Selecting a note in the TalkExplorer resurfaces its contextual anchors: the corresponding pointer traces and design elements reappear within the main canvas, situating the designer in the original context of their thought and supporting reflection or continuation of interrupted reasoning.

4.2.5 **TalkReminders**: *Resurfacing Related Prior Notes (DP4).* In addition to manual browsing, the system proactively resurfaces relevant prior notes through TalkReminders.

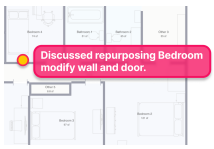


Figure 14

When the user's current verbalization relates to earlier concerns, previously created TalkNotes briefly reappear on the canvas, highlighted and summarized next to their anchors. This lightweight mechanism helps jog memory and connect ongoing reasoning with past decisions without requiring explicit search or navigation.

4.3 Implementation Details

We developed PointAloud as a web-based prototype, following a client–server architecture. The front end was implemented in *React*

[1]Action suggestion buttons are generated per TalkNote but remain non-functional in our prototype, serving solely as probes.

with *Three.js* for 3D rendering and *TLDraw* for the 2D sketching canvas. The back end, built with *Node.js* and *Express*, manages data flow between transcription services *(Deepgram Nova-3)* and large language models (LLMs) APIs *(GPT-4o, Gemini 2.5 Pro)*. For further implementation details, see Appendix Section A.1.

5 User Study

To better understand possible benefits and limitations of pointer-centric AI-supported think-aloud computing interactions, we conducted a within-subject remote user study aimed at providing insights into these research questions:

RQ1a *What are the key differences between working with real-time text-based transcription only and PointAloud?*
RQ1b *Does PointAloud incentivize people to think aloud more?*
RQ2 *How do people think aloud and work with PointAloud?*
RQ3 *What are users' perceived benefits and challenges for working with PointAloud?*

These research questions aim to understand different aspects of the PointAloud interaction suite, collectively unpacking hypotheses on how users respond to these four aspects: *think-aloud computing*, *pointer-adjacent ambient displays*, *pointer-attention context*, and *AI-driven support features*. Questions *RQ1a* and *RQ2a* largely explore the effect of pointer-adjacent ambient displays on think-aloud computing. While *RQ2* equally touches on each of these interaction suite concepts to understand how they interplay to support the CAD task scenario. Finally, *RQ3* tends to lean toward insights on AI-driven support as participants would tend to associate those features as the benefits of the system.

For RQ1a and RQ1b, we decided to compare PointAloud interactions with a modified (limited) version of the PointAloud system, where instead of the `TalkPointer` and `TalkNote` features, the live user's transcription appears in the left side panel as text (optionally see Appendix Figure 20). This limited, low distraction system (baseline) aims to mimic the contemporary form of live transcription system UI (for example, in Zoom or Teams) – the baseline provides support for think-aloud capture, without the additional embedded user interfaces of PointAloud.

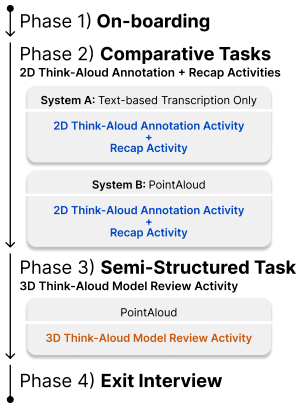


Figure 15: Process diagram of the four-phase study procedure.

5.1 Participants

We recruited 12 participants *(7 self-identified as female and 5 as male, aged 23–49 years, $M = 30.8$, $SD = 6.7$)* with professional backgrounds in architecture and interior design. Participants had between 2 and 19 years of professional experience *($M = 6.9$, $SD = 4.5$)*, see Appendix Table 2. Recruitment was conducted via snowball sampling and professional online user forums. Eligibility was determined through a screening form, requiring a minimum of two years of professional experience in architecture or interior design, as well as at least two years of CAD experience and fluent English-level proficiency. All participants signed an IRB-approved consent form prior to the first study session and were compensated with a $125 gift card upon completion.

5.2 Procedure and Tasks

The 90-minute-long remote study was structured into four phases:

Phase 1) On-Boarding (20–30 min): At the beginning of the session, the facilitator welcomed participants, introduced the study, and ensured that consent was signed. Participants then completed a short pre-task survey and watched a three-minute tutorial video demonstrating PointAloud's core functionalities. Afterward, they performed a five-minute guided hands-on trial to familiarize themselves with the prototype interface and features.

Phase 2) Comparative Tasks (2 × 20 min): In the comparative phase, participants completed two design tasks, alternating between a baseline version of the prototype (limited to live text-based transcription) and the full version of PointAloud. Each task was paired with a distinct design brief (similar to the task described in 4.1.1), with the order of briefs counterbalanced across participants to mitigate learning effects. These design briefs were intentionally framed to elicit early-stage design reasoning, requiring participants to repurpose an apartment layout while balancing functional requirements with spatial constraints and emerging client needs. For each task, participants first worked for ten minutes on a **2D floor plan annotation activity** while verbalizing their thoughts aloud, followed by a five-minute **recap activity** in which they summarized their early design decisions, open questions, and issues to flag as if preparing for a client meeting. After each task, participants completed an attitudinal post-task questionnaire.

Phase 3) Semi-structured Task (5 min): In the third phase, participants were asked to continue working on their previous design brief, this time by reviewing a spatial model of the floor in the 3D model view (similar to the task described in 4.1.2). They could only view and explore a model of the apartment (without making changes or annotations) to analyze the existing space and reflect on interior design considerations. Participants worked for five minutes using the full prototype system while continuing to think aloud. This task was designed to encourage spontaneous engagement with the prototype's features in a spatial 3D interface.

Phase 4) Exit Interview (15 min): In the final phase, participants took part in a semi-structured interview. They reflected on their experience with PointAloud, comparing it to their usual tools and workflows and discussing its effectiveness in supporting and harnessing think-aloud practices. Participants provided feedback on strengths, limitations, and opportunities for integrating such a system into professional design work.

5.3 Collected Data, Measures, and Analysis

Across the study, we collected the following data:

- Video, screen, audio recordings, and machine-generated transcripts of all task sessions (phase 2 and 3)
- System interaction log data of all task sessions (phase 2 and 3)
- Post-task surveys data (phase 2)
- Audio recordings and machine-generated transcripts of the exit interviews (phase 4)

To compare the baseline (text-based transcript-only system) and PointAloud (Q1a), we analyzed the post-task surveys from phase 2 that probed participants' perceived thinking-aloud support, task support (including relevance of system suggestions), process awareness, and cognitive load/distraction on a 6-point Likert scale. We applied the Wilcoxon signed-rank test to assess statistical significance and calculated 95% confidence intervals for mean differences via bootstrapping with 10,000 replications using R [62]. This approach has been suggested for similar data and studies [47, 77].

To assess to what extent PointAloud incentivized participants to think aloud (Q1b), we compared participants' words-per-minute (WPM) across the baseline and the PointAloud condition [2]. We then conducted paired comparisons using the Wilcoxon signed-rank test and paired t-test to assess differences in WPM between conditions.

To answer how people think aloud and work with PointAloud (Q2), we conducted a video interaction analysis [4] of the video recordings collected in the PointAloud-supported activities in phase 2 and 3. Using a custom-built coding tool, we reviewed the video recordings alongside the system interaction log files for each session and coded participants' interactions, such as their verbal responses to system suggestions. From this data, we then created timeline visualizations for each session using ggplot2 [72].

Finally, to investigate users' perceived benefits and challenges of working with PointAloud (Q3), we conducted a reflexive thematic analysis [6] of the interview transcripts. We followed an iterative inductive coding process and generated themes through affinity diagramming using Dovetail [12].

6 Study Findings

6.1 Key differences: Live Text-based Transcription vs PointAloud (RQ1a, RQ1b)

In the two comparative tasks (phase 2), participants worked on apartment redesign briefs that asked them to annotate a 2D floor plan and articulate early zoning ideas while thinking aloud. They then completed a recap activity, summarizing key design decisions, open questions, and concerns as if preparing for a client meeting using the annotated floor plan and the think-aloud text-based transcript or TalkNotes, respectively. Here, we report on the observed differences in participants' processes when working with conventional live text-based transcription (baseline) versus PointAloud. Optionally, we included screenshots in Appendix A.2 to give an impression of the variety of participant-created floor plans using PointAloud.

[2]Transcripts were tokenized using spaCy [27]. We counted tokens that were not punctuation or whitespace and contained at least one alphabetic character. Common English contractions were merged and treated as single words.

6.1.1 ***Questionnaire.*** In the comparative post-task surveys, participants consistently rated PointAloud higher than the baseline text-based transcription system across both task phases (Figure 16). During the **floorplan annotation activity**, participants reported a significant improvement in **relevance of system suggestions** (*"…provided relevant suggestions helpful for the task"*, $MD = 1.83$, $p = 0.007$), **process awareness** (*"…helped keep track of the design process"*, $MD = 1.25$, $p = 0.007$), and **task support** (*"…supported completing the annotation task"*, $MD = 1.08$, $p = 0.01$). They also noted stronger **thinking-aloud support** (*"…supported thinking aloud during the task"*, $MD = 0.83$, $p = 0.04$), while ratings for **cognitive load / distraction** showed no significant difference ($MD = 0.08$, $p = 0.887$). While the baseline lacks the PointAloud features, it provides an opportunity to meaningfully compare the cognitive load and distraction between the two conditions. Overall, it is expected that participants would prefer PointAloud for these tasks compared to the baseline, however additional benefits from PointAloud did not increase effort or attention when compared to the baseline.

During the **client recap activity**, PointAloud was valued for aiding **memory and reflection** (*"…helped remembering and reflecting on what I had done earlier"*, $MD = 1.50$, $p = 0.007$), **summarization and communication** (*"…made it easier to summarize and communicate design decisions"*, $MD = 1.25$, $p = 0.011$), and **issue identification** (*"…helped identifying issues or challenges"*, $MD = 1.08$, $p = 0.018$). Participants also perceived improved **organization of open questions and assumptions**, though this effect was marginal ($MD = 0.92$, $p = 0.085$). During the brief recap activity, participants preferred the AI-supported documentation features of PointAloud compared to the text-based baseline. On reflection, the baseline might be improved with AI support such as chat or generative search powered by Retrieval-Augmented Generation (RAG) [39]. Future work should take this into consideration when evaluating think-aloud systems.

Across both phases, participants expressed feeling **more productive than with their typical tools and workflows** when using PointAloud ($MD = 0.83$, $p = 0.02$).

6.1.2 ***Observations.*** **Differences During Think-Aloud Activity.** With the text-based transcription, participants spoke aloud while their speech appeared in the left side panel as text. However, the transcript was never actively visited and used during this task activity. In contrast, with PointAloud, participants' utterance stream was displayed via TalkText and TalkViz near their cursor while periodically forming anchored TalkNotes on the canvas and side panel. While TalkTips frequently appeared next to the cursor, participants' responses varied widely. Some rarely engaged with them, while others actively used them to elaborate their reasoning and to guide ongoing design decisions. Similarly, TalkNotes themselves functioned differently across participants—ranging from passive documentation to active use as a springboard for reflection and further exploration. These varied engagement patterns with TalkNotes and TalkTips during think-aloud are analyzed in more detail in Section 6.2.

No Difference in Extent of Verbalization (RQ1b). For the *annotation* think-aloud task activities, we compared words-per-minute

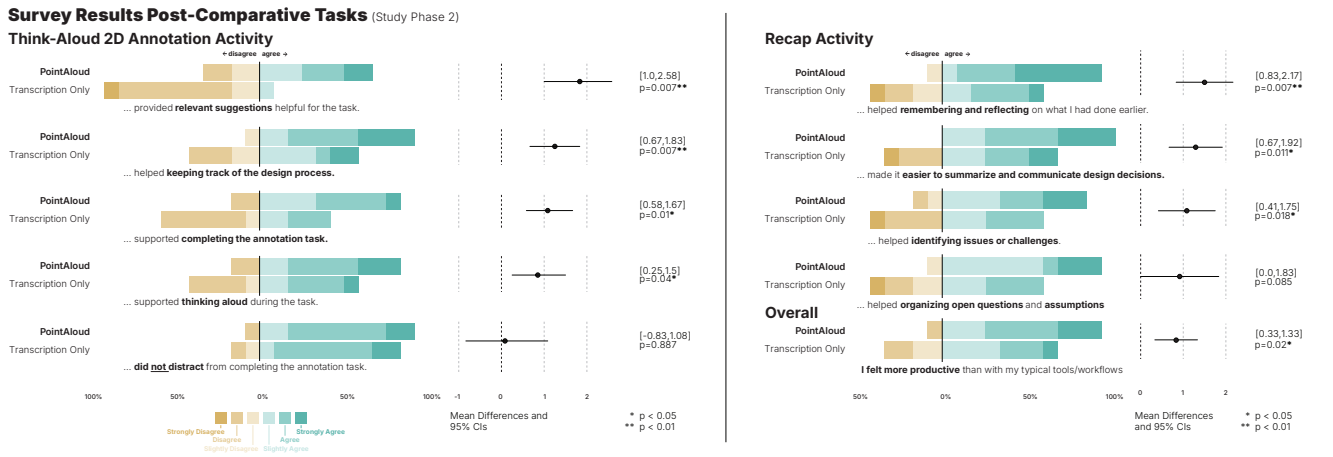


Figure 16: Participants' responses when rating the 6-point Likert statements for annotation and recap activities completed with PointAloud and text-based live transcription only (baseline), ranked from largest to smallest effects; Dots show the mean difference of PointAloud compared to Text-based Transcription Only; Bars are the 95% CIs calculated with the studentized bootstrap method.

(WPM) across the two conditions to assess whether PointAloud incentivized more verbalization. Differences were participant-specific: some (e.g., P07, P10) spoke considerably more with PointAloud, while others (e.g., P04, P11) spoke less (for further details see also Appendix Table 3). On average, the difference across conditions was negligible (Mean = 1.23 WPM, SD = 17.46). Neither a paired t-test ($t = -0.245$, $p = .81$) nor a Wilcoxon signed-rank test ($W = 34$, $p = .73$) indicated a significant effect. This suggests that PointAloud did not uniformly increase the quantity of verbalization, though it shifted how participants engaged with their speech as design material. However, because participants were explicitly instructed to think aloud in both conditions, our study design does not isolate whether PointAloud itself prompted additional verbalization.

Extreme Differences During Recap Activity. When completing their *recap activity* in which participants summarized their earlier design process as if preparing for a client meeting, engagement diverged strongly between conditions. With text-based transcription alone, no participant made substantive use of the transcript (only P07 briefly skimmed the transcript for a few seconds). In contrast, `TalkNotes` in PointAloud provided an externalized memory that participants engaged with to varying degrees. As shown in Figure 17, **we identified three usage patterns during the recap activity** with PointAloud in regard to users' engagement with `TalkNotes` and the `TalkExplorer`'s category filters:

- ***Light Recap Users*** (e.g., P01, P04, P05, P09) who made minimal or no use of PointAloud (`TalkNotes` and category filter).
- ***Iterative Recap Users*** (e.g., P03, P06, P07, P08) who selectively revisited and filtered `TalkNotes` to organize their summaries.
- ***Power Recap Users*** (e.g., P02, P10, P11, P12) who engaged extensively, strategically filtering and cross-referencing notes to construct detailed recaps.

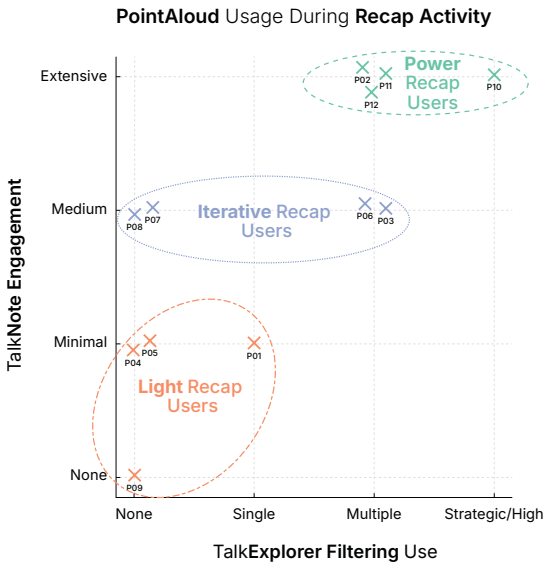


Figure 17: Patterns of `TalkNote` engagement and `TalkExplorer` filtering use during the recap activity, distinguishing *Light Users*, *Iterative Users*, and *Power Users*.

These differences highlight how PointAloud not only captured verbalizations but also supported many users in retrospectively reflecting on their design process and structuring their documented ideas for communicating their process.

Table 1: Overview of the results of the PointAloud-supported think-aloud activities (2D annotation and 3D review activities combined); reporting session duration and process statistics for TalkNotes and TalkTips interactions. Columns with conditional coloring indicate relative intensity within the measure. Final column indicates user engagement pattern: *Note Explorer* (●), *Tip-driven Elaborator* (●), *Heavy Integrator* (●+●), *Documentation-only User* (see Section 6.2). * *Note: In P02's first activity,* TalkNote *creation was periodically interrupted by a browser plug-in conflict.*

ID	Activity Duration (mm:ss)	# TalkNotes Created	# TalkNotes Merged	# User-Checked TalkNotes	# TalkTips Shown	# TalkTip Responses	Engagement Pattern
P01	15:18	28	13	11	63	2	● Note Explorer
P02*	14:29	35	13	0	56	3	Documentation-only
P03	15:44	40	26	0	83	1	Documentation-only
P04	15:23	57	27	6	94	0	● Note Explorer
P05	14:07	55	30	0	91	2	Documentation-only
P06	15:22	55	27	4	73	24	●+● Heavy Integrator
P07	15:06	60	32	0	70	3	Documentation-only
P08	14:56	58	41	0	76	1	Documentation-only
P09	15:19	42	24	0	92	1	Documentation-only
P10	15:23	63	36	6	78	12	●+● Heavy Integrator
P11	13:37	46	22	0	62	1	Documentation-only
P12	15:20	65	40	1	64	7	● Tip-driven Elaborator
Mean	*15:01*	*50.3*	*27.6*	*2.3*	*75.2*	*4.8*	

6.2 Observed Think-Aloud Workflows Using PointAloud (RQ2)

During the two PointAloud-supported think-aloud activities, participants repurposed an apartment for childcare by first annotating the floor plan in 2D (*phase 2*), then reviewing and reflecting on their design in the 3D viewer (*phase 3*).

To analyze how participants interacted with PointAloud's features throughout the tasks, we aggregated system and user interaction event counts of both think-aloud activities from phase 1 and phase 2 (see Table 1) and visually compared all sessions' timeline visualizations (optionally, see Figure 19 in the Appendix). Overall, **we identified four different engagement patterns during the think-aloud activities** across participant sessions, according to their interactions with the TalkNotes and TalkTip features during the think-aloud planning activity— *Note Explorers* (●), *Tip-driven Elaborators* (●), *Heavy Integrators* (●+●), and *Documentation-only Users* (see Figure 18)—which we unpack below:

Varied Engagement Patterns with* TalkNotes*: From Background Documentation to Active Think-Aloud Support. TalkNotes were generated automatically as participants verbalized, and the system created between 28 and 65 TalkNotes per session (M = 50.3, SD = 11.9) across both 2D/3D activities (see sequences of squares ❐/■ in Figure 18 and Figure 19). However, participants varied in whether they *engaged with these notes during the think-aloud activities*: Most participants refrained from exploring the captured notes, while some occasionally revisited them by clicking/hovering (see sequences of triangles ▲ in Figure 18, Figure 19). For example, P01, P04, P06, and P10 repeatedly inspected their TalkNotes in the 2D and 3D activities (between 11 and 4 inspected TalkNotes), while most others left the accumulating notes untouched. In the case of P04, after annotating the floor plan for about eight minutes, the participant paused to scroll through the TalkNotes in the side panel, explicitly revisiting earlier thoughts (*"just reviewing my notes here..."*). This review prompted further elaboration, as they considered follow-up questions about zoning, occupancy, sound, and parking (see Figure 18). Overall, these observations highlight two broad workflows: *Documentation-only users* who treated TalkNotes primarily as an *automatic background capture for documentation purposes* while thinking aloud, and those who acted as *note explorers* (●), actively integrating TalkNotes into their reasoning process while thinking aloud. The variability in engagement patterns hints toward user preferences in how participants designed while thinking aloud. Identifying the rationale behind their preferences was not part of our initial hypotheses. These varied engagement patterns emerged from questions on how users would think aloud and work with PointAloud (RQ2). However, the identification of these patterns provides food for thought for further explorations into think-aloud computing and work process documentation.

Varied Response Patterns to* TalkTips*: From Ignoring to Guided Exploration. TalkTips are brief, pointer-attached prompts delivered both during pauses and in response to users' verbalizations. Across both 2D/3D tasks, participants saw between 56 and 94 TalkTips (M = 75.2, SD = 12.8; see Table 1). Response behavior varied widely—from no responses (e.g., P04) to sustained engagement (e.g., P06 with 24 responses)—and often shifted with task context.

For example, in the 3D model review, P12 read and replied to multiple prompts while scanning the main room. When prompted

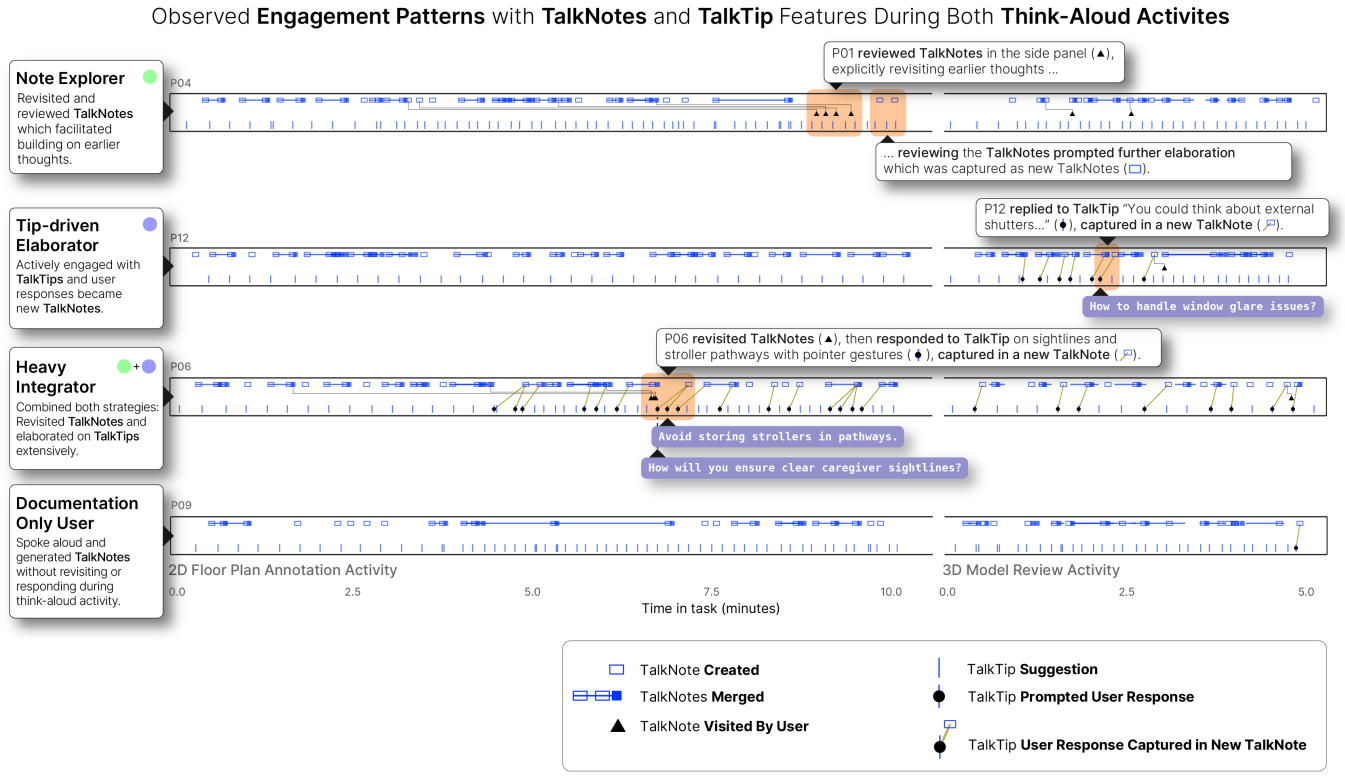


Figure 18: Timeline visualizations of interaction events, illustrating four engagement patterns with TalkNotes and TalkTips during the 2D floor plan annotation and 3D model review activities: *Note Explorer*, *Tip-driven Elaborator*, *Heavy Integrator*, and *Documentation-only User*.

about window glare (TalkTip: "How to handle window glare issues?"), P12 immediately proposed actionable options: *"You could think about external shutters, which will regulate temperature as well..."* (see Figure 18 and optionally Appendix Figure 32b). This pattern—brief system cue, situated inspection, concise verbal elaboration—recurred as P12 moved through adjacent areas of the plan.

An even more intensive workflow appeared with P06, who repeatedly used TalkTips to steer their reasoning in rapid succession. Around minute 7, after a short review of their TalkNotes (*"looking back at everything I've said..."*), a prompt surfaced: "How will you ensure clear caregiver sightlines?" P06 responded while pointer-gesturing over the plan, articulating a rationale for room layout and corridor openness. Moments later, a safety prompt ("Avoid storing strollers in pathways") led P06 to clarify the circulation zones verbally (*"strollers to the corner, away from the main path"*), which the system captured as a new TalkNote summarizing the design intent as *"Caregiver access prioritized; strollers redirected from pathways to corners."* (see Figure 18, optionally Appendix Figure 26a). Here, the sequence of TalkTips, pointer-driven think-aloud activity, and TalkNotes formed a tight loop of prompt→reflection→decision capture.

Taken together, we observed two characteristic workflows in response to TalkTips: *tip-driven elaborators* (●), who frequently replied and extended their thinking aloud, and *documentation-only users*, who largely ignored system prompts. A third pattern, *heavy integrators* (●+●), cycled between reviewing TalkNotes and answering TalkTips to progress their design reasoning, using prompts to probe trade-offs and then consolidating decisions into new notes.

Contrasting TalkTip Engagement Across 2D Annotation and 3D Review Activities. Participants responded to more TalkTips in the 2D annotation task activity (phase 2) than in the shorter 3D review task (phase 3) when looking at absolute counts ($M = 2.8$ vs. $M = 2.0$ responses per session). However, after normalizing for task duration, the rate of engagement was slightly higher in 3D ($M = 0.39$ responses/min) compared to 2D ($M = 0.27$ responses/min), though with a skewed distribution: in 2D most participants engaged a little, while in 3D a few engaged intensively but many did not respond at all (see also Appendix Figure 19). Interview reflections provide a lens on this discrepancy: several participants reported feeling more attentive to TalkTips in the 3D review task, since they were not simultaneously sketching and could devote more cognitive bandwidth to responding. As P02 explained, *"Having to do the annotations [in 2D] distracted me from noticing those [TalkTip] bubbles. But in the 3D space, because the annotation was taken away completely, those things were more noticeable. So I was looking at the wall and then I saw the [TalkTip]* 'Hey, think about

the structure. Load bearing.' *And I was like 'Oh, yeah. Let's think about the load bearing.' So that was really helpful [...] and only noticeable when I didn't have to worry about annotation."*

Together, these findings reveal a tension between observed behavior and perceived experience: while absolute logs show similar engagement in 2D annotation task and 3D review task, participants experienced TalkTips as more salient in 3D without additional annotations.

TalkReminders as Subtle but Ineffective Cues. TalkReminders were designed to proactively resurface relevant prior notes: when a participant's current verbalization related to earlier concerns, previously created TalkNotes briefly reappeared on the canvas, highlighted and summarized next to their anchors. Although such reminders appeared in the logs (between 26 and 48 times during participant sessions), our video analysis did not reveal participants visibly responding to them. Participants either ignored or did not consciously register these resurfaced notes, suggesting that the reminders were too subtle to shape behavior or that participants noticed them but did not perceive them as actionable in the moment. For example, P12 desired to have notes re-surfaced based on their current process: *"I wanted to [...] go back to my notes [...] to a certain thought [...] especially when you're on a fast [...] iterative processes."* However, they found browsing through TalkNotes to sidetrack their train-of-thought: *"instead of, like, helping me find exactly what I was going for, I was just [...] met with all of these other issues and [Talk]notes."* Despite a preference to have relevant TalkNotes re-surfaced, this testimonial indicates that TalkReminders went unseen by P12.

6.3 Users' Perceived Benefits and Challenges For Working With PointAloud (RQ3)

We clustered the perceived benefits and challenges for users working with PointAloud into three themes: *Supporting Thinking Aloud*, *Supporting Design Process*, and *Supporting Human-AI Co-Creation.*

6.3.1 *Supporting Thinking Aloud.*

Users appreciated thinking aloud for fostering a more reflective practice and how the system helped them stay in the flow. Overall, participants saw value in verbalizing their thoughts during the tasks, with several framing verbalization as a welcomed reflective design practice: *"I don't usually [. . .] talk out loud, [but] this kind of speaking [my thoughts] helps me to reflect on my own thinking"* (P03). Many participants also reported that speaking while working would lead to a more organized design process, as P04 described: *"It forces you to organize your thoughts as you're going through the space."* (P04)

Participants also described that verbalization was experienced as focus-enhancing rather than intrusive and that real-time capture allowed them to keep moving and stay in the "flow" without stopping to take notes, with TalkNotes acting as lightweight "traces" of thinking, as P12 described: *"[It was] capturing these little snippets of thought as I was pretty much [. . .] free flowing. The fact that the notes are being taken automatically, it's a cool thing. That might [. . .] lead you to focus more actually and not lose your train of thought, which is quite precious."*

Users felt motivated by the system's ambient nature that created a sense of being "listened to.' Participants highlighted how the unobtrusive ambient transcription display of the TalkPointer and TalkNotes created a sense of being "listened to," which motivated them to keep verbalizing, as P11 explained: *"This tool really helps me to stay focused [...] it's transcribing what I was talking about [...] I don't feel like it was distracting [...] because I was treating it as some ambient activity [and] it's not occupying that much space of the screen, so you don't have to really pay attention to what is going on there. But I do have a sense, 'oh, something is happening there.' And I know that this computer is, like, listening to me. So [...] it motivated me to keep talking."*

Others compared the system to a conversational partner that helped articulate and surface ideas that might otherwise remain unspoken: *"It makes sense because it's like working with another person [...] you have to express [your thoughts] out loud [because otherwise] some of the ideas never come to the surface."* (P10) Participants also reflected on how the TalkText feature influenced their verbalization. For several, seeing the live transcript next to the cursor prompted clearer articulation. Others also mentioned that having the live transcript feed next to their cursor would reduce context switching by not having to *"look on the other side of the screen to see the transcription."* (P01)

While* TalkText *signaled verbalization activity, participants largely ignored it while speaking. Overall, participants felt motivated by the system to think aloud, but many also reported that while TalkText signaled activity, they rarely paid attention to the transcribed words and activity visualization, as P04 reflected: *"It gave a cue that it was capturing what I was saying [...] but because it wasn't readily identifiable [...] I kind of ignored it."* For a few, TalkText also introduced anxiety around accuracy and the risk of being misunderstood. Together, these reflections highlight how TalkText shaped awareness of being recorded and encouraged clarity, while also revealing challenges around limited engagement with the displayed text and worries about potential transcription errors.

Thinking aloud also introduced a learning curve and felt impractical for some. At the same time, a few participants also described a learning curve and initial hesitations around thinking aloud. For example, P06 mentioned: *"Thinking out loud is a little bit of a vulnerable thing to do [...] I was afraid a little bit that [it] would judge you [...] but then I saw [...] it's beneficial to think out loud."* Lastly, P08 also doubted the practicality of thinking outside the study setting: *"Thinking aloud doesn't really work in an office environment because you cannot [...] talk out loud [near] other people [and] if I'm working alone, I'm not gonna talk aloud [...] it takes energy [...] it wouldn't be as practical and functional, but it can work in some settings."*

6.3.2 *Supporting Design Process.*

Users valued the tool's engaging support for externalizing, structuring and resurfacing their fleeting thoughts. Participants emphasized how PointAloud made their design process feel more engaging, deliberate and manageable. By externalizing fleeting thoughts, the tool helped structure what might otherwise remain chaotic, as P11 put it: *"I've never imagined designing something*

this way [...] Because this makes [the] design process [...] much more engaging for me as a designer. Also [...] one struggle for me while doing design is [...] thinking about this complex problem from multiple dimensions. You sometimes [get lost] in your own thinking. And this tool is really helpful, in terms of documenting what is still a chaos in my head, but it's somehow jogging down everything in a very efficient way. And what is even cooler is that it helps you to kind of organize your thinking" (P11).

The system was also described as a promising way to archive and resurface thoughts for later recall, as P05 explained: *"It's like your thoughts are not going in the way [...] it's just putting them in a book or a file [...] that will help you when you keep thinking and designing, and you come back to this zone later."*

Many participants valued how the system captured their situational reasoning and made design intent accessible for later reference. In particular, TalkNotes were seen as more effective than having the raw text transcript, as P02 reflected: *"I didn't really use the [text only] transcript that much. But [...] the talk notes were actually quite helpful. So I did forget a lot, [...] So recording that and going back to those talk notes was actually quite helpful to record what my sort of thought process was."*

Users saw benefits in linking TalkNotes to 2D/3D spatial contexts. Participants highlighted the value of linking TalkNotes to specific elements in the floor plan or 3D model, which supported later recall and continuity. Many imagined TalkNotes as a new form of note-taking, combining verbalization with spatial, pointer-centric context, as P04 put it: *"It is an interesting idea [...] verbalizing your thought process for the design and how it's highlighting the different areas."* Frequently, participants valued the novelty and convenience of automated transcription and cursor location to place notes directly in relevant spaces by *"moving my cursor and just talking, it would add the notes to the area that I'm talking about."* (P08) Linking notes with floor plans and 3D views preserved context, making it easier to retrieve details and make sense of past work: *"Another thing that is helpful for me is the visual connection between the floor plans, the 3D view and the notes. [...] Sometimes we remember we said something, but we don't remember the exact details of the context [...] And [it would be impossible documenting my process in detail how] this tool can provide."* (P11)

Interestingly, some participants found notes easier to interpret in 3D, where design discussions like wall changes were more clearly tied to spatial context than in 2D, as mentioned by P02: *"This is actually very helpful in 3D. Because, [...] it's a lot easier for me to understand or see these points and where they're related to in the 3D model compared to the 2D canvas."*

Participants highlighted challenges of automatic pointer-attention alignment. A recurring reported friction was that the pointer would not always represent where participants' attention was directed, as P04 reflected: *"The tricky part is [...] when I'm going through the thought process, I'm moving the mouse a lot as I am thinking [...] so it may be difficult to pinpoint my notes to where they're relevant in that particular space."* P05 also mentioned that their awareness of the pointer capture could feel distracting if misaligned, worrying it might create inaccurate TalkNotes: *"Because I know it links what I say to where the mouse is, it sometimes distracts me when I maybe forget [to move] my mouse pointer to here [...] and I'm talking about another space. [This] would create confusion."* This recurring misalignment calls to attention one of the pitfalls of using pointer location data as a proxy for user attention, future work could consider ways to filter out irrelevant pointer location data based on the ongoing user context. For example, what the user is talking about should align with what they are pointing at.

Users valued TalkNote grouping and category filtering for easily navigating their documented process. Participants emphasized that the automatic grouping of TalkNotes into TalkThreads and their semantic classification into process categories supported clearer retrieval of their design process, as P06 reflected: *"I think something that helped me [was] that it classified [the TalkNotes] so I was able to check: 'What is the process that I went through?' [...] So it doesn't overwhelm you when you filter that. I usually like when things are sorted out [and] grouped together. I like the fact that it does it for you."* (P06) Categorization and filtering were also seen as valuable for navigating large sets of notes and directly revisiting specific moments: *"I wanted to [...] check which was one of the problem areas that I kinda forgot [...] it was really nice to be able to click on 'problem' and see exactly what I needed."* (P12) At the same time, participants expressed a desire for customization, noting that predefined labels might not always match their personal workflow, and also for the ability to search for TalkNotes and arrange them more flexibly outside of the list view.

Participants saw great potential in TalkNotes for also supporting human—human collaboration. Participants also frequently envisioned the system not only as a personal tool but also as a shared resource for collaborative design. They described how TalkNotes could allow teammates to follow each other's reasoning without requiring extensive explanation. P10 expanded on how passing along notes with design intent was seen as particularly valuable: *"I like that there's added data that you can pass along. So it's not just your first pass on the layout design, but also your thoughts before making it. So it could provide some insights to the next person [...] That's good information because sometimes [...] you don't know who else designed this or what they were thinking of. That's very helpful [...] like a preview [...] from another designer's perspective."*

Participants also noted potential information boundaries for certain collaborations. While TalkNotes were valuable for internal teams, external consultants might require filtered or tailored views, as P02 elaborated: *"Internally, it could work well. [But] working with separate consultants [...] might be a bit confusing. They might not want all of that information. So if there was an ability to maybe give cues and filter [...] that might be quite helpful."*

6.3.3 *Supporting Human-AI Co-Creation*.

From a "thought recorder" to co-creative AI support. Participants described how PointAloud's context-aware suggestions shifted the system from a passive recorder of verbalized thoughts to a virtual collaborator that could help externalize ideas and seed design moves. As P10 noted, *"There are times where you actually need some feedback [...] it's helpful for the design process [...] you are able to, like, bounce ideas out with someone else."* Others emphasized how novel it felt for a design tool to actively engage, as P06 reflected: *"I have never seen a design tool that [...] tries to understand what I'm saying, and that was really cool to watch, actually."* Similarly, P05 highlighted the

practical value of the TalkNotes' AI-generated Action Suggestions, observing that *"You can see what the AI got from your speech and it gives you more suggestions [...] Maybe these are links for some furniture, similar projects, or some permit or code pages [...] That's always so helpful."*

TalkTips can aid designers' thinking without taking over. When aligned with participants' current focus, TalkTips acted as lightweight thinking supports—providing nudges, references, and reminders that helped them sustain momentum without feeling overridden. For instance, P06 described, *"the prompts were very impactful [...] it made me think about what I actually need to think about. Because I often deviate, but it kept putting me back on track. So that was really cool [...] and I felt like whenever I would stop talking, the prompt would come up. And that was something I liked."*

Overall mixed feelings about proactive system suggestions: value vs. distraction when misaligned. While many valued the system's proactive nudges, a few participants also described distraction when timing, persistence, or placement did not match their current focus. P09 summarized this tension: *"The prompts [were] distracting [...] because it kept on saying something that I was not thinking about at that point. But it might also be helpful to read because it was prompting me to think about the lighting quality in this space. So it is distracting, but at the same time, needed [...] I have mixed feelings about that."* Some struggled to notice or process TalkTip suggestions while working or found their transient presentation too brief, as P04 reflected, *"I didn't really read it while I was working. My eyes did notice this bar piece that was showing up [...] but I didn't really comprehend the text that was being written."* Participants also flagged cursor-adjacent pop-ups as interruptive, as P12 shared: *"Talking out loud reinforces and is helpful [...] but not necessarily the notes popping by my cursor [...] I found that rather distracting [...] breaking my train of thought."*

7 Discussion

In the following sections, we reflect on the lessons from designing, implementing, and studying *PointAloud*, focusing on how pointer-centric interactions can *incentivize Think-Aloud Computing*, *provide real-time feedback through pointer-ambient displays*, enable richer forms of *process-aware human–AI co-creation*, and support *design process documentation*. We also discuss how the PointAloud interaction suite contributes a transferable approach for other domains, such as writing in a text editor interface and visual data analysis via a computational notebook. Each subsection highlights challenges, design considerations, and open questions for advancing pointer-centric think-aloud computing into everyday workflows.

7.1 Incentivizing Think-Aloud Computing

Building on the original notion of Think-Aloud Computing [38], a central question of our work is whether interaction techniques such as PointAloud meaningfully incentivize people to verbalize their thoughts during design tasks. Our study provides a nuanced answer. Quantitative analysis showed no significant difference in words-per-minute (WPM) between PointAloud and the baseline live transcription condition (see Section 6.1.2). This suggests that PointAloud did not uniformly increase the amount of verbalization. However, because participants were explicitly instructed to think aloud in both conditions, our study design does not isolate whether PointAloud itself prompted additional articulation. Future dedicated studies are needed to assess this effect better.

While the volume of speech did not change, qualitative findings indicate that PointAloud influenced how participants experienced and valued verbalization assisted by PointAloud (Section 6.3). Many highlighted benefits such as a sense of being "listened to," greater clarity of articulation, and the usefulness of externalizing fleeting thoughts for later reflection and recap. In some cases, participants also engaged actively with TalkTips in rapid succession, producing extended chains of verbalized reasoning that were immediately documented as TalkNotes (Section 6.2). Thus, while overall speech quantity remained stable between conditions, PointAloud appeared to incentivize specific forms of verbalization. Feedback from participants indicates a change from a monologue to a dialogue with the system. The participants thought aloud, and the system responded via real-time documentation (TalkNotes) and AI-generated suggestions (TalkNote Actions, TalkTips).

At the same time, thinking aloud also presented a novel practice for most participants. Participants quickly adapted and saw value in thinking aloud, while some described initial hesitation, and some questioned its practicality in office settings where speaking aloud can feel socially inappropriate (Section 6.3.1). This echoes challenges discussed in prior work on integrating think-aloud methods into everyday computing contexts [38], underscoring that adoption requires not only functional features but also attention to comfort and situational fit.

In other domains, such as writing, the need to incentivize users to think aloud may be greater, because verbally expressing thoughts while simultaneously writing text may be unfamiliar or cognitively demanding. However, other parts of the writing process may feel more natural, such as verbalizing fleeting ideas during the ideation phase of writing. Also, more problem-solving-oriented tasks, such as visual data analysis, might lend themselves to more frequent verbalization, particularly when analysts ask questions aloud about the data.

Design Considerations. First, incentivizing users to think aloud requires anchoring verbalizations to visible and useful outcomes, such as immediate feedback, design suggestions, or process documentation. Second, systems must make the benefits of speaking aloud transparent to users in the moment. For example, PointAloud visibly demonstrates how the user's verbalizations (TalkText and TalkVis) translate into actionable TalkNotes or trigger relevant TalkTips. Finally, designers should account for the learnability of think-aloud computing: it may require gradual onboarding, scaffolds that ease initial discomfort, and longitudinal use before its full value may be realized.

Open Questions. Future research should examine in which task contexts think-aloud computing most effectively encourages verbalization. Without instructing participants to verbalize their thoughts, do they think aloud solely to gain the benefits of the system? How does sustained use over weeks or months shift both the quantity and quality of articulation? Addressing these questions will be critical to understanding whether interaction suites like PointAloud can sustainably embed think-aloud practices into everyday design workflows.

7.2 Designing Pointer-Adjacent Ambient Displays for Real-time Feedback

Our findings show that pointer-adjacent ambient displays, such as TalkText and TalkVis, provided valuable reassurance that speech was being captured without distracting from design tasks. Survey results revealed no increase in perceived distraction compared to baseline transcription ($p = 0.887$, see Section 6.1.1), suggesting that embedding capture indicators around the pointer offers a low-friction way to build awareness and trust. However, participants often reported only glancing at these elements rather than continuously monitoring them, indicating that such feedback must strike a balance: frequent enough to confirm capture, but lightweight and glanceable to avoid shifting attention toward evaluating transcription quality.

Other ambient mechanisms were less effective. TalkReminders were rarely acted upon (Section 6.2), likely because their subtle resurfacing did not call users to action. By contrast, TalkTips were more salient and sometimes prompted rapid cycles of reflection and decision-making, though their effectiveness varied with task context. Notably, participants experienced TalkTips as more noticeable in 3D model review tasks than in 2D annotation, a difference they attributed to having more cognitive bandwidth when not simultaneously sketching. This underscores how task load mediates the salience of pointer-adjacent interfaces.

Overall, these observations extend prior work on peripheral and ambient displays (e.g., *Ambient Help* [49], *SidePoint* [41], *FeedQUAC* [42]), which emphasize embedding contextual information into users' ongoing workflows. PointAloud contributes to this space by demonstrating how pointer-adjacent cues can reassure users about process capture while also serving as proactive verbalization and thinking supports.

The pointer-adjacent interface featured in PointAloud would need to be reconsidered for other user interface paradigms. For example, in a text editor interface, used in both writing and data analysis, the user's attention is digitally represented at differing moments by the pointer and the text cursor. Identifying which of these simultaneous locations best represents the user's attention is an open question. However, capturing the user's focus may be improved in other domains that introduce selection interaction paradigms. In writing, users could select text and speak aloud about a desired change. Also, when working in visual data analysis, users could directly select data points of interest in interactive charts. We hypothesize that selection, coupled with think-aloud verbalizations, could increase clarity on the user's underlying intent.

Design Considerations. Ambient pointer displays should remain subtle and glanceable, updating frequently enough to provide reassurance without overloading the user. Proactive cues must be tuned to task context and cognitive load, while secondary cues such as TalkReminders require more careful design to make them more noticeable and actionable.

Open Questions. Future work should investigate how to adapt the salience of pointer-ambient feedback dynamically to different activity phases and workload levels. What forms of lightweight error correction (e.g., voice-based commands) can help users manage transcription without leaving the flow of work? And how can such displays scale to richer system-suggested actions that operate on the design itself (e.g., auto-complete) without becoming overwhelming?

7.3 Towards Process-Aware Human-AI Co-Creation

Beyond documentation, a central promise of think-aloud computing is to enable more context-aware forms of human–AI co-creation. By capturing users' ongoing verbalized reasoning, PointAloud provides AI systems with semantically rich data about design intent that can be used to deliver more aligned and situated assistance. Our study highlights this potential: participants valued TalkTips when they connected directly to their current reasoning, and described the system as shifting from a passive recorder toward an active collaborator (see Section 6.3). At the same time, when prompts were misaligned with the task at hand, participants experienced them as distracting (Section 6.3). This underscores that effective AI support requires more than task-domain knowledge—it must also be attuned to where users are in their process.

PointAloud contributes here by coupling speech capture with additional process signals such as pointer traces and design context, which provide further cues about what users are attending to in the workspace. These multimodal traces offer opportunities for anticipating user needs, for example, by tailoring suggestions based on what element is being inspected or which concern has just been verbalized. Related research on workflow capture and mixed-initiative systems has long emphasized the importance of balancing proactive support with user control [16, 30], and our findings extend this by showing how verbalization data can anchor system initiative in the *evolving reasoning* of the user.

An opportunity for improving human-AI co-creation lies in using think-aloud data not only for semantic context but also for inferring underlying cognitive and metacognitive user states. Prior work in learning analytics has shown that self-regulated learning phases—such as planning, enacting, and evaluating—can be detected from think-aloud protocols [3, 5]. Applied to think-aloud computing design workflows, similar approaches could allow systems to detect whether a user is in early ideation, evaluating trade-offs, or is unfamiliar with a certain software feature, and to adapt proactive support accordingly. Complementary work also suggests that cursor movements can act as behavioral indicators of users' attention and cognitive states [9], further pointing to how multimodal traces might enrich models of user state.

Think-aloud computing has the opportunity to enrich the user context for AI support systems. In writing, a tight editing loop could be powered by pointer-centric think-aloud interactions where the user selects text, verbalizes intended changes, and the LLM-driven support generates recommendations based on that context. Furthermore, think-aloud data attached to sections of writing could preserve the writer's intention. This metadata about the user's intent provides rationale and commentary that could enrich downstream LLM-recommended changes to the text. In visual data analysis, visualization recommendation has improved based on LLM-supported pipelines driven by natural language question and answering [44, 74]. However, introducing PointAloud concepts could

move data analysis toward mixed-initiative systems where the analysis process is an ongoing iteration based on user intention. In this new model, AI agents could operate in the background based on the user's spoken hypotheses and appraisals.

Design Considerations. To be effective supporters, AI agents must be not only context-aware but also *process-aware*, responding differently to exploratory reasoning versus evaluative reflection. Capturing speech alongside pointer and interaction signals offers a foundation for such adaptive support, but systems must remain transparent about how inferences are drawn to sustain trust.

Open Questions. Future work should examine how reliably think-aloud transcripts and multimodal traces can be used to infer metacognitive states in real-world design contexts. How should AI agents calibrate the timing and content of suggestions to different phases of work without becoming intrusive? What ethical implications arise when systems infer or attempt to influence users' cognitive states? Exploring these questions will be critical for realizing the full potential of think-aloud computing in human–AI co-creation.

7.4 PointAloud Interfaces as a Solution for Design Process Documentation

A core contribution of PointAloud lies in demonstrating a new approach for documenting design processes. Participants valued the system for helping externalize, structure, and archive their fleeting reasoning in ways that traditional live transcription did not (Section 6.3). `TalkNotes` enabled participants to resurface earlier decisions, link ideas directly to spatial context in 2D and 3D, and retrieve information through grouping and category filtering. These features made design processes more tangible and navigable, supporting reflection and recap while reducing the risk of losing important rationales. In this way, PointAloud instantiates a transferable method for capturing tacit decision-making processes that are often left undocumented.

Beyond individual use, the findings also point to the potential of PointAloud as a collaborative documentation tool. Participants imagined `TalkNotes` as artifacts that could be shared with colleagues to explain their reasoning without extensive verbal walkthroughs (Section 6.3). By situating documentation directly in the design workspace, PointAloud opens pathways for process knowledge to travel across users, tasks, and applications. This aligns with prior systems for workflow capture and retrieval [23, 64, 69], and extends them by linking verbalized rationales with spatial pointer context. Such an approach could connect across diverse tools (e.g., CAD models, project management systems, BIM environments) to create richer, cross-application process histories.

Taken together, these findings suggest that PointAloud contributes a novel way of documenting and sharing design processes, moving beyond capturing *"what was done"* toward preserving *"why it was done."* This dual emphasis on action and rationale reflects a broader need in professional design practice to surface tacit knowledge and support continuity across time and collaborators.

The research field of visual data analysis has a rich history in tracking user interactions and visualization provenance [63, 75]. Think-aloud capture introduces low-effort user rationale that can be attached to the existing provenance user interfaces and interactions in visual data analysis research. Additionally, the PointAloud AI-driven method for documenting work processes may be helpful in organizing and structuring streams of data analysis and resulting insights for future recall.

Design Considerations. Systems for process documentation should tightly couple reasoning with context—whether spatial, temporal, or semantic—to make captured knowledge useful both for individual reflection and for collaborative coordination. Lightweight retrieval mechanisms, such as grouping and filtering, can reduce overwhelm and support targeted reuse.

Open Questions. Future work should also investigate alternative representations of captured processes, such as chronological timelines that complement spatial anchoring. How might such documentation scale to complex, multi-user projects, and what forms of representation best support reuse across heterogeneous tools and workflows? Addressing these questions will help extend pointer-centric documentation into broader ecosystems of collaborative design practice.

8 Conclusion

Think-Aloud Computing offers the potential to capture rich contextual insights into users' evolving intentions, struggles, and decision-making in real time. Yet, existing approaches face challenges: users often lack awareness of what is being captured, are not sufficiently encouraged to verbalize their thoughts, and may miss or be disrupted by system feedback. Moreover, thinking aloud must feel worthwhile by yielding meaningful assistance. To address these challenges, we introduced *PointAloud*, a suite of AI-driven pointer-centric interactions designed for in-the-moment verbalization encouragement, low-distraction system feedback, and contextually rich process documentation alongside proactive AI assistance. We instantiated these techniques in the *PointAloud System*, a CAD application for annotating 2D floor plans and inspecting 3D architectural models, allowing us to explore pointer-centric think-aloud computing in a concrete design context. Our user study with 12 participants demonstrates how pointer-centric think-aloud support can facilitate documentation and enrich human–AI co-creation. Building on these findings, we outline design considerations for future pointer-centric and AI-supported Think-Aloud Computing workflows, including strategies for incentivizing verbalization, designing pointer-ambient displays, enabling more process-aware human–AI co-creation, and embedding documentation seamlessly within users' ongoing workflows. While our study focused on architectural design, the PointAloud interaction suite illustrates a transferable design pattern that could inform other AI-assisted creative and knowledge work scenarios, offering HCI researchers and practitioners a novel interaction approach for integrating AI into in-the-moment user reasoning, documenting work processes, and enabling richer forms of human-AI co-creation.

Acknowledgments

We thank all study participants and the reviewers for their constructive feedback on the paper. The prototype was implemented with assistance from GitHub Copilot, with all generated code reviewed, edited, and tested by the authors. Figures 1 and 2 contain image elements (illustrations of people) that were generated with

ChatGPT. Several figures in this paper include screenshots of 2D floor plans and corresponding 3D models sourced from Polycam community content, used under the Creative Commons Attribution 4.0 (CC BY 4.0) license.

References

[1] Mark Ashdown and Thad Starner. 2005. Method and System for Displaying Context-Sensitive Information Using a Hybrid Cursor. Patent No. US20050088410A1.

[2] Paul Ayres and John Sweller. 2005. The Split-Attention Principle in Multimedia Learning. In *The Cambridge Handbook of Multimedia Learning* (1 ed.), Richard Mayer (Ed.). Cambridge University Press, 135–146. doi:10.1017/CBO9780511816819.009

[3] Maria Bannert, Peter Reimann, and Christoph Sonnenberg. 2014. Process Mining Techniques for Analysing Patterns and Strategies in Students' Self-Regulated Learning. *Metacognition and Learning* 9, 2 (Aug. 2014), 161–185. doi:10.1007/s11409-013-9107-6

[4] Eric P.S. Baumer and Bill Tomlinson. 2011. Comparing Activity Theory with Distributed Cognition for Video Analysis: Beyond "Kicking the Tires". In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. Association for Computing Machinery, New York, NY, USA, 133–142. doi:10.1145/1978942.1978962

[5] Conrad Borchers, Jiayi Zhang, Ryan S. Baker, and Vincent Aleven. 2024. Using Think-Aloud Data to Understand Relations between Self-Regulation Cycle Characteristics and Student Performance in Intelligent Tutoring Systems. In *Proceedings of the 14th Learning Analytics and Knowledge Conference*. 529–539. doi:10.1145/3636555.3636911 arXiv:2312.05675 [cs]

[6] Virginia Braun and Victoria Clarke. 2019. Reflecting on Reflexive Thematic Analysis. *Qualitative Research in Sport, Exercise and Health* 11, 4 (Aug. 2019), 589–597. doi:10.1080/2159676X.2019.1628806

[7] Senthil Chandrasegaran, Chris Bryan, Hidekazu Shidara, Tung-Yen Chuang, and Kwan-Liu Ma. 2019. TalkTraces: Real-Time Capture and Visualization of Verbal Content in Meetings. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, Glasgow Scotland Uk, 1–14. doi:10.1145/3290605.3300807

[8] Urška Demšar and Arzu Çöltekin. 2017. Quantifying Gaze and Mouse Interactions on Spatial Visual Interfaces with a New Movement Analytics Methodology. *PLOS ONE* 12, 8 (Aug. 2017), e0181818. doi:10.1371/journal.pone.0181818

[9] M.R. Dias da Silva and M. Postma. 2020. Wandering Minds, Wandering Mice: Computer Mouse Tracking as a Method to Detect Mind Wandering. *Computers in Human Behavior* 112 (Nov. 2020), 106453. doi:10.1016/j.chb.2020.106453

[10] Graham Dove, Nicolai Brodersen Hansen, and Kim Halskov. 2016. An Argument For Design Space Reflection. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction (NordiCHI '16)*. Association for Computing Machinery, New York, NY, USA, 1–10. doi:10.1145/2971485.2971528

[11] Ashley Dover, G. Michael Poor, Darren Guinness, and Alvin Jude. 2016. Improving Gestural Interaction With Augmented Cursors. In *Proceedings of the 2016 Symposium on Spatial User Interaction (SUI '16)*. Association for Computing Machinery, New York, NY, USA, 135–138. doi:10.1145/2983310.2985765

[12] Dovetail. 2025. Customer Insights Hub — Dovetail. https://dovetail.com/.

[13] Claudia Eckert and Jean-François Boujut. 2003. The Role of Objects in Design Co-Operation: Communication through Physical or Virtual Objects. *Computer Supported Cooperative Work (CSCW)* 12, 2 (June 2003), 145–151. doi:10.1023/A:1023954726209

[14] Figma. 2025. Cursor Chat in Figma Design. https://help.figma.com/hc/en-us/articles/4403130802199-Use-cursor-chat-in-Figma-Design.

[15] Adam Fourney, Richard Mann, and Michael Terry. 2011. Query-Feature Graphs: Bridging User Vocabulary and System Functionality. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. ACM, Santa Barbara California USA, 207–216. doi:10.1145/2047196.2047224

[16] C. Ailie Fraser, Mira Dontcheva, Holger Winnemöller, Sheryl Ehrlich, and Scott Klemmer. 2016. DiscoverySpace: Suggesting Actions in Complex Software. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*. ACM, Brisbane QLD Australia, 1221–1232. doi:10.1145/2901790.2901849

[17] C. Ailie Fraser, Tricia J. Ngoon, Mira Dontcheva, and Scott Klemmer. 2019. RePlay: Contextually Presenting Learning Videos Across Software Applications. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, Glasgow Scotland Uk, 1–13. doi:10.1145/3290605.3300527

[18] Frederic Gmeiner, Kaitao Luo, Ye Wang, Kenneth Holstein, and Nikolas Martelaro. 2025. Exploring the Potential of Metacognitive Support Agents for Human-AI Co-Creation. In *Proceedings of the 2025 ACM Designing Interactive Systems Conference (DIS '25)*. Association for Computing Machinery, New York, NY, USA, 1244–1269. doi:10.1145/3715336.3735785

[19] Frederic Gmeiner, Humphrey Yang, Lining Yao, Kenneth Holstein, and Nikolas Martelaro. 2023. Exploring Challenges and Opportunities to Support Designers in Learning to Co-create with AI-based Manufacturing Design Tools. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, Hamburg Germany, 1–20. doi:10.1145/3544548.3580999

[20] Jun Gong, Fraser Anderson, George Fitzmaurice, and Tovi Grossman. 2019. Instrumenting and Analyzing Fabrication Activities, Users, and Expertise. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–14. doi:10.1145/3290605.3300554

[21] Floraine Grabler, Maneesh Agrawala, Wilmot Li, Mira Dontcheva, and Takeo Igarashi. 2009. Generating photo manipulation tutorials by demonstration. *ACM Trans. Graph.* 28, 3, Article 66 (July 2009), 9 pages. doi:10.1145/1531326.1531372

[22] Tovi Grossman and George Fitzmaurice. 2010. ToolClips: An Investigation of Contextual Video Assistance for Functionality Understanding. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Atlanta Georgia USA, 1515–1524. doi:10.1145/1753326.1753552

[23] Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2010. Chronicle: Capture, Exploration, and Playback of Document Workflow Histories. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology (UIST '10)*. Association for Computing Machinery, New York, NY, USA, 143–152. doi:10.1145/1866029.1866054

[24] Thomas Gruber, John Boose, Catherine Baudin, and Jay Weber. 1991. Design Rationale Capture as Knowledge Acquisition: Tradeoffs in the Design of Interactive Tools. In *Machine Learning Proceedings 1991*. Elsevier, 3–12. doi:10.1016/B978-1-55860-200-7.50006-4

[25] Carl Gutwin, Saul Greenberg, and Mark Roseman. 1996. Workspace Awareness in Real-Time Distributed Groupware: Framework, Widgets, and Evaluation. In *People and Computers XI*, Martina Angela Sasse, R. Jim Cunningham, and Russel L. Winder (Eds.). Springer London, London, 281–298. doi:10.1007/978-1-4471-3588-3_18

[26] Onur Hisarciklilar and Jean-François Boujut. 2009. An Annotation Model to Reduce Ambiguity in Design Communication. *Research in Engineering Design* 20, 3 (Sept. 2009), 171–184. doi:10.1007/s00163-009-0073-6

[27] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python. *Zenodo* (2020). doi:10.5281/zenodo.1212303

[28] John Horner and Michael E. Atwood. 2006. Design Rationale: The Rationale and the Barriers. In *Proceedings of the 4th Nordic Conference on Human-computer Interaction: Changing Roles*. ACM, Oslo Norway, 341–350. doi:10.1145/1182475.1182511

[29] Amber Horvath, Andrew Macvean, and Brad A Myers. 2024. Meta-Manager: A Tool for Collecting and Exploring Meta Information about Code. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI '24)*. Association for Computing Machinery, New York, NY, USA, 1–17. doi:10.1145/3613904.3642676

[30] Eric Horvitz. 1999. Principles of Mixed-Initiative User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems the CHI Is the Limit - CHI '99*. ACM Press, Pittsburgh, Pennsylvania, United States, 159–166. doi:10.1145/302979.303030

[31] Eric J. Horvitz, John S. Breese, David Heckerman, David Hovel, and Koos Rommelse. 2013. The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users. doi:10.48550/arXiv.1301.7385 arXiv:1301.7385 [cs]

[32] 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*. ACM, Austin Texas USA, 1341–1350. doi:10.1145/2207676.2208591

[33] Anam Ahmad Khan, Joshua Newn, James Bailey, and Eduardo Velloso. 2022. Integrating Gaze and Speech for Enabling Implicit Interactions. In *CHI Conference on Human Factors in Computing Systems*. ACM, New Orleans LA USA, 1–14. doi:10.1145/3491102.3502134

[34] Anam Ahmad Khan, Joshua Newn, Ryan M. Kelly, Namrata Srivastava, James Bailey, and Eduardo Velloso. 2021. GAVIN: Gaze-Assisted Voice-Based Implicit Note-taking. *ACM Transactions on Computer-Human Interaction* 28, 4 (Aug. 2021), 1–32. doi:10.1145/3453988

[35] Kimia Kiani, Parmit K. Chilana, Andrea Bunt, Tovi Grossman, and George Fitzmaurice. 2020. "I Would Just Ask Someone": Learning Feature-Rich Design Software in the Modern Workplace. In *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. 1–10. doi:10.1109/VL/HCC50065.2020.9127288

[36] Jon Kolko. 2010. Sensemaking and Framing: A Theoretical Reflection on Perspective in Design Synthesis.

[37] Nicholas Kong, Tovi Grossman, Björn Hartmann, Maneesh Agrawala, and George Fitzmaurice. 2012. Delta: A Tool for Representing and Comparing Workflows. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. Association for Computing Machinery, New York, NY, USA, 1027–1036. doi:10.1145/2207676.2208549

[38] Rebecca Krosnick, Fraser Anderson, Justin Matejka, Steve Oney, Walter S. Lasecki, Tovi Grossman, and George Fitzmaurice. 2021. Think-Aloud Computing: Supporting Rich and Low-Effort Knowledge Capture. In *Proceedings of the 2021 CHI*

Conference on Human Factors in Computing Systems. ACM, Yokohama Japan, 1–13. doi:10.1145/3411764.3445066

[39] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (Nips '20)*. Curran Associates Inc., Red Hook, NY, USA, Article 793.

[40] Daniel J. Liebling and Susan T. Dumais. 2014. Gaze and Mouse Coordination in Everyday Work. *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication* (Sept. 2014), 1141–1150. doi:10.1145/2638728.2641692

[41] Yefeng Liu, Darren Edge, and Koji Yatani. 2013. SidePoint: a peripheral knowledge panel for presentation slide authoring. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 681–684. doi:10.1145/2470654.2470750

[42] Tao Long, Kendra Wannamaker, Jo Vermeulen, George Fitzmaurice, and Justin Matejka. 2025. FeedQUAC: Quick Unobtrusive AI-Generated Commentary. doi:10.48550/arXiv.2504.16416 arXiv:2504.16416 [cs]

[43] Zhicong Lu, Seongkook Heo, and Daniel J. Wigdor. 2018. StreamWiki: Enabling Viewers of Knowledge Sharing Live Streams to Collaboratively Generate Archival Documentation for Effective In-Stream and Post Hoc Learning. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW (Nov. 2018), 112:1–112:26. doi:10.1145/3274381

[44] Jock Mackinlay, Pat Hanrahan, and Chris Stolte. 2007. Show Me: Automatic Presentation for Visual Analysis. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1137–1144. doi:10.1109/TVCG.2007.70594

[45] Karthik Mahadevan, Qian Zhou, George Fitzmaurice, Tovi Grossman, and Fraser Anderson. 2023. Tesseract: Querying Spatial Design Recordings by Manipulating Worlds in Miniature. 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 460, 16 pages. doi:10.1145/3544548.3580876

[46] Shareen Mahmud, Jessalyn Alvina, Parmit K. Chilana, Andrea Bunt, and Joanna McGrenere. 2020. Learning Through Exploration: How Children, Adults, and Older Adults Interact with a New Feature-Rich Application. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–14. doi:10.1145/3313831.3376414

[47] Damien Masson, Sylvain Malacria, Géry Casiez, and Daniel Vogel. 2023. Statslator: Interactive Translation of NHST and Estimation Statistics Reporting Styles in Scientific Documents. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. ACM, San Francisco CA USA, 1–14. doi:10.1145/3586183.3606762

[48] Damien Masson, Jo Vermeulen, George Fitzmaurice, and Justin Matejka. 2022. Supercharging Trial-and-Error for Learning Complex Software Applications. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems (CHI '22)*. Association for Computing Machinery, New York, NY, USA. doi:10.1145/3491102.3501895

[49] Justin Matejka, Tovi Grossman, and George Fitzmaurice. 2011. Ambient help. 10 pages. doi:10.1145/1978942.1979349

[50] Richard E. Mayer and Logan Fiorella. 2014. Principles for Reducing Extraneous Processing in Multimedia Learning: Coherence, Signaling, Redundancy, Spatial Contiguity, and Temporal Contiguity Principles. In *The Cambridge Handbook of Multimedia Learning* (2 ed.), Richard Mayer (Ed.). Cambridge University Press, Cambridge, 279–315. doi:10.1017/CBO9781139547369.015

[51] Miro. 2024. Miro. https://miro.com/.

[52] Brad A. Myers. 2024. *Pick, Click, Flick!: The Story of Interaction Techniques* (1 ed.). ACM, New York, NY, USA. doi:10.1145/3617448

[53] Alok Mysore and Philip J. Guo. 2017. Torta: Generating Mixed-Media GUI and Command-Line App Tutorials Using Operating-System-Wide Activity Tracing. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. ACM, Québec City QC Canada, 703–714. doi:10.1145/3126594.3126628

[54] Tricia J. Ngoon, Joy O Kim, and Scott Klemmer. 2021. Shöwn: Adaptive Conceptual Guidance Aids Example Use in Creative Tasks. In *Proceedings of the 2021 ACM Designing Interactive Systems Conference* (Virtual Event, USA) *(DIS '21)*. Association for Computing Machinery, New York, NY, USA, 1834–1845. doi:10.1145/3461778.3462072

[55] Fred Paas and John Sweller. 2014. Implications of Cognitive Load Theory for Multimedia Learning. In *The Cambridge Handbook of Multimedia Learning* (2 ed.), Richard Mayer (Ed.). Cambridge University Press, Cambridge, 27–42. doi:10.1017/CBO9781139547369.004

[56] Srishti Palani, Yingyi Zhou, Sheldon Zhu, and Steven P. Dow. 2022. InterWeave: Presenting Search Suggestions in Context Scaffolds Information Search and Synthesis. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. ACM, Bend OR USA, 1–16. doi:10.1145/3526113.3545696

[57] Soya Park, Amy X. Zhang, and David R. Karger. 2018. Post-Literate Programming: Linking Discussion and Code in Software Development Teams. In *Adjunct Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (UIST '18 Adjunct)*. Association for Computing Machinery, New York, NY, USA, 51–53. doi:10.1145/3266037.3266098

[58] Alexander R Payne, Beryl Plimmer, Andrew McDaid, Andrew Luxton-Reilly, and T Claire Davies. 2016. Expansion Cursor: A Zoom Lens That Can Be Voluntarily Activated by the User at Every Individual Click. In *Proceedings of the 28th Australian Conference on Computer-Human Interaction - OzCHI '16*. ACM Press, Launceston, Tasmania, Australia, 81–90. doi:10.1145/3010915.3010942

[59] Peter Pirolli and Stuart Card. 2005. The Sensemaking Process and Leverage Points for Analyst Technology as Identified through Cognitive Task Analysis. In *Proceedings of the International Conference on Intelligence Analysis*, Vol. 5. McLean, VA, USA, 2–4.

[60] Zachary Pousman and John Stasko. 2006. A Taxonomy of Ambient Information Systems: Four Patterns of Design. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '06)*. Association for Computing Machinery, New York, NY, USA, 67–74. doi:10.1145/1133265.1133277

[61] Thanawit Prasongpongchai, Pat Pataranutaporn, Monchai Lertsutthiwong, and Pattie Maes. 2025. Talk to the Hand: An LLM-powered Chatbot with Visual Pointer as Proactive Companion for On-Screen Tasks. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, 1–16. doi:10.1145/3706598.3715579

[62] R Core Team. 2024. *R: A Language and Environment for Statistical Computing*. Vienna, Austria.

[63] Eric D. Ragan, Alex Endert, Jibonananda Sanyal, and Jian Chen. 2016. Characterizing Provenance in Visualization and Data Analysis: An Organizational Framework of Provenance Types and Purposes. *IEEE Transactions on Visualization and Computer Graphics* 22, 1 (2016), 31–40. doi:10.1109/TVCG.2015.2467551

[64] Søren Rasmussen, Jeanette Falk Olesen, and Kim Halskov. 2019. Co-Notate: Exploring Real-time Annotations to Capture Situational Design Knowledge. In *Proceedings of the 2019 on Designing Interactive Systems Conference*. ACM, San Diego CA USA, 161–172. doi:10.1145/3322276.3322310

[65] I. M. M. J. Reymen. 2003. Research on Design Reflection: Overview and Directions. *DS 31: Proceedings of ICED 03, the 14th International Conference on Engineering Design, Stockholm* (2003), 33–34 (exec.summ.), full paper no. DS31_1148FPB.

[66] Donald A. Schön. 1983. *The Reflective Practitioner: How Professionals Think in Action*. Basic Books, New York.

[67] Yang Shi, Chris Bryan, Sridatt Bhamidipati, Ying Zhao, Yaoxue Zhang, and Kwan-Liu Ma. 2018. MeetingVis: Visual Narratives to Assist in Recalling Meeting Context and Content. *IEEE Transactions on Visualization and Computer Graphics* 24, 6 (June 2018), 1918–1929. doi:10.1109/TVCG.2018.2816203

[68] Balasaravanan Thoravi Kumaravel, Cuong Nguyen, Stephen DiVerdi, and Björn Hartmann. 2019. TutoriVR: A Video-Based Tutorial System for Design Applications in Virtual Reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, Glasgow Scotland Uk, 1–12. doi:10.1145/3290605.3300514

[69] April Yi Wang, Zihan Wu, Christopher Brooks, and Steve Oney. 2020. Callisto: Capturing the "Why" by Connecting Conversations with Computational Narratives. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3313831.3376740

[70] Bryan Wang, Meng Yu Yang, and Tovi Grossman. 2021. Soloist: Generating Mixed-Initiative Tutorials from Existing Guitar Instructional Videos Through Audio Processing. 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 98, 14 pages. doi:10.1145/3411764.3445162

[71] Christopher Wickens. 2021. Attention: Theory, Principles, Models and Applications. *International Journal of Human–Computer Interaction* 37, 5 (March 2021), 403–417. doi:10.1080/10447318.2021.1874741

[72] Hadley Wickham. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.

[73] Jacob O. Wobbrock, James Fogarty, Shih-Yen (Sean) Liu, Shunichi Kimuro, and Susumu Harada. 2009. The Angle Mouse: Target-Agnostic Dynamic Gain Adjustment Based on Angular Deviation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Boston MA USA, 1401–1410. doi:10.1145/1518701.1518912

[74] Aoyu Wu, Yun Wang, Xinhuan Shu, Dominik Moritz, Weiwei Cui, Haidong Zhang, Dongmei Zhang, and Huamin Qu. 2022. AI4VIS: Survey on Artificial Intelligence Approaches for Data Visualization. *IEEE Transactions on Visualization and Computer Graphics* 28, 12 (2022), 5049–5070. doi:10.1109/TVCG.2021.3099002

[75] Kai Xu, Alvitta Ottley, Conny Walchshofer, Marc Streit, Remco Chang, and John Wenskovitch. 2020. Survey on the Analysis of User Interactions and Visualization Provenance. *Computer Graphics Forum* 39, 3 (2020), 757–783. doi:10.1111/cgf.14035

[76] Qian Zhou, George Fitzmaurice, and Fraser Anderson. 2022. In-Depth Mouse: Integrating Desktop Mouse into Virtual Reality. In *CHI Conference on Human Factors in Computing Systems*. ACM, New Orleans LA USA, 1–17. doi:10.1145/3491102.3501884

[77] Yaqian Zhu and John Kolassa. 2018. Assessing and Comparing the Accuracy of Various Bootstrap Methods. *Communications in Statistics - Simulation and Computation* 47, 8 (Sept. 2018), 2436–2453. doi:10.1080/03610918.2017.1348516

A Additional Materials

Table 2: Overview of study participants.

ID	Age	Gender	Role	Years of Prof. Exp.
P01	31	Female	Interior designer	2
P02	27	Male	Architect	6
P03	26	Male	Assistant Architect	6
P04	49	Male	Faculty / Facilities Director / Licensed Architect	19
P05	34	Male	Head of Architecture Design Department	11
P06	23	Female	Construction Management Intern / Architect	6
P07	30	Female	Interior Designer	3
P08	30	Female	Architectural Designer	4
P09	25	Female	Real Estate Intern / Architect	7
P10	29	Female	Exhibition Design Intern	5
P11	30	Female	Interior Designer	5
P12	35	Male	Artist Studio Designer	9

Table 3: Per-participant words per minute (WPM) across baseline (Task A) and PointAloud (Task B).

PID	Task A WPM (Baseline)	Task B WPM (PointAloud)	Diff (B–A)
P01	70.5	55.7	-14.8
P02	79.5	85.1	+5.6
P03	65.4	76.3	+10.9
P04	110.8	82.4	-28.4
P05	74.9	70.3	-4.6
P06	100.2	96.1	-4.1
P07	73.7	105.4	+31.7
P08	49.5	65.1	+15.6
P09	34.6	40.8	+6.2
P10	61.9	82.6	+20.7
P11	102.7	81.9	-20.8
P12	87.1	83.9	-3.2
Mean Diff		+1.23 (SD = 17.46)	

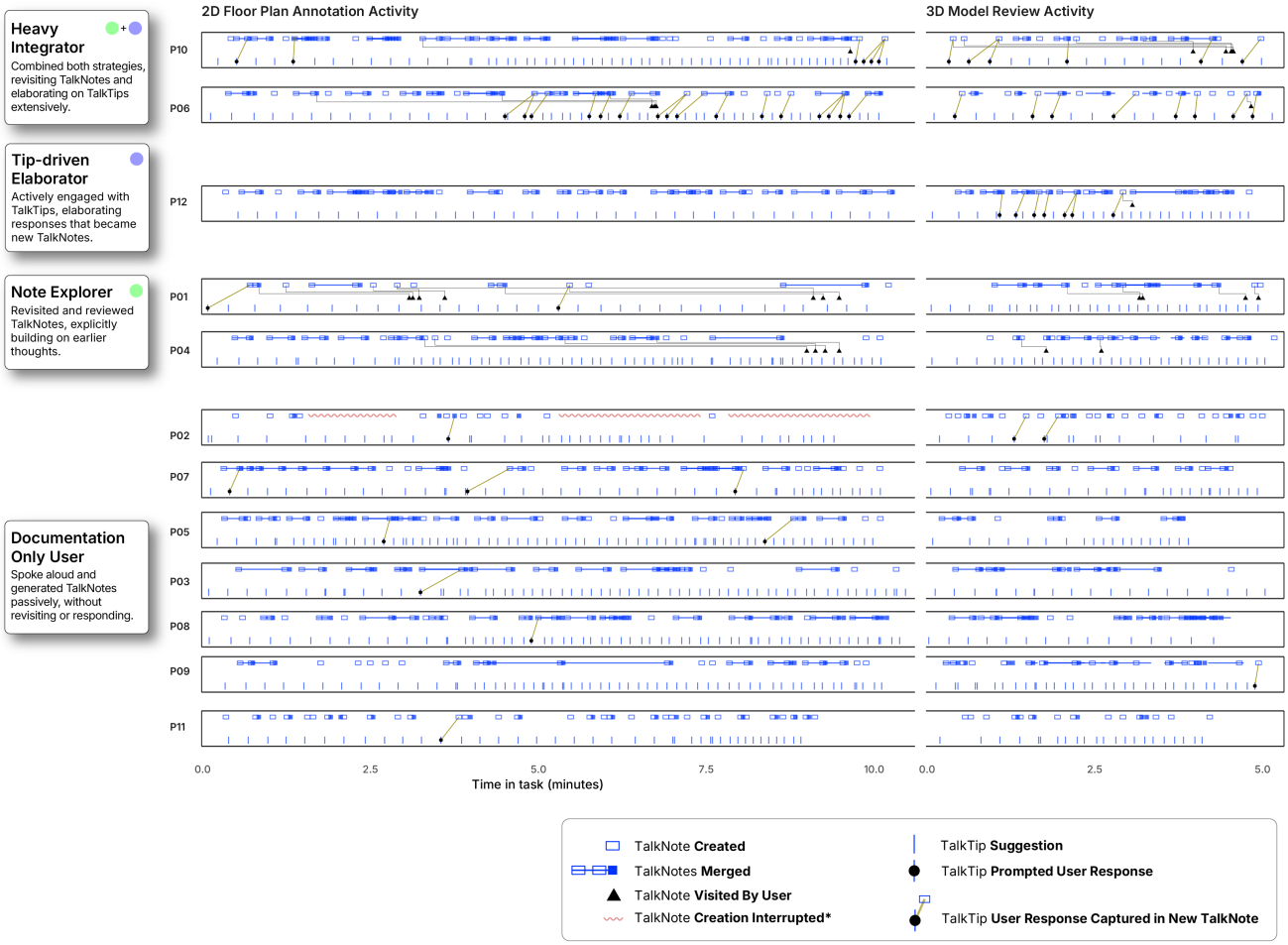


Figure 19: Timeline visualizations of interaction events of all user sessions, clustered into four engagement patterns with TalkNotes and TalkTips during the 2D floor plan annotation and 3D model review activities: *Note Explorer*, *Tip-driven Elaborator*, *Heavy Integrator*, and *Documentation-only User*. **Note: In P02's first activity, TalkNote creation was periodically interrupted by a browser plug-in conflict.*

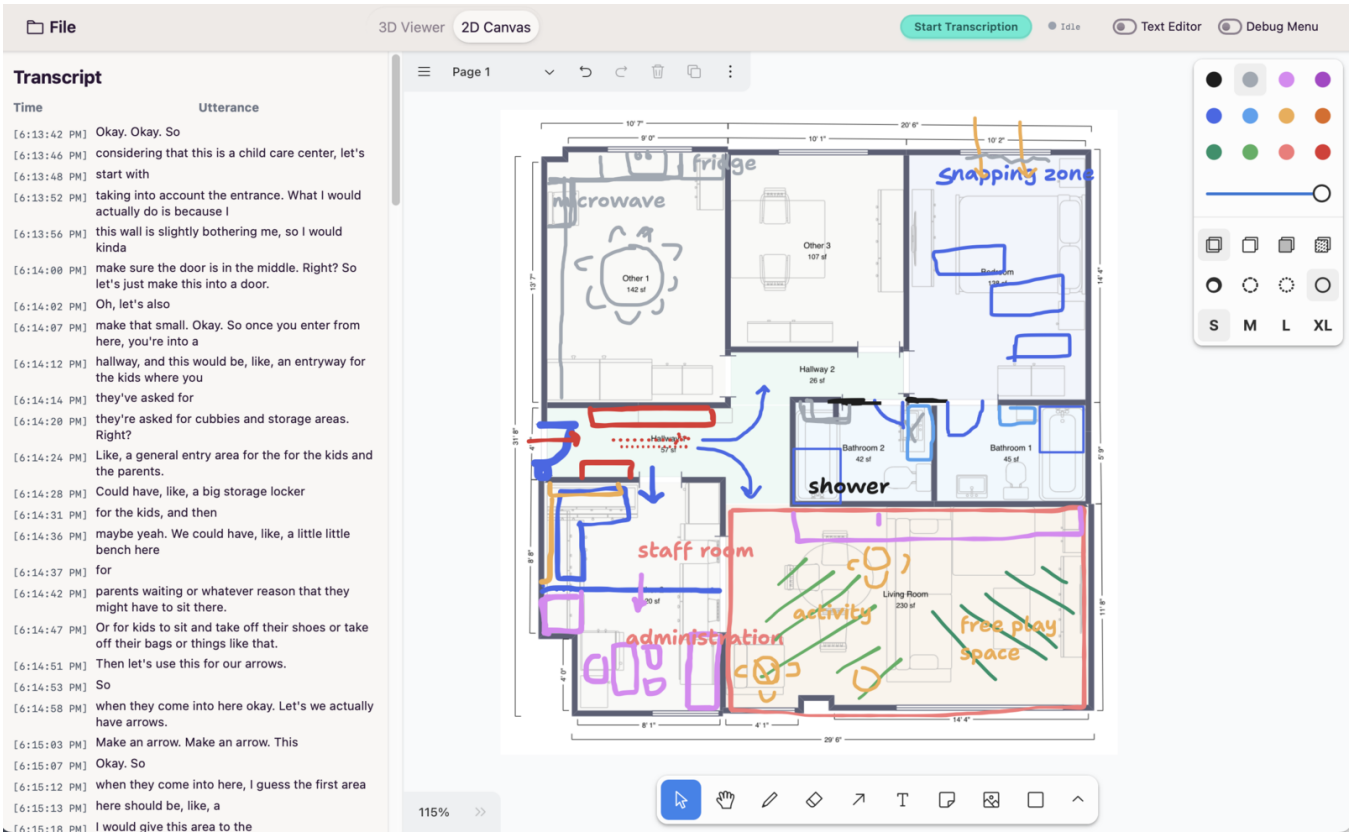


Figure 20: Screenshot of the PointAloud system with the text-based live transcription in the left side panel serving as the baseline used by participants in the comparative tasks (phase 2).

A.1 Implementation Details

A.1.1 ***Speech Transcription and Language Models.*** User speech is captured via the browser's microphone input and streamed to a commercial transcription service (*Deepgram Nova-3*). Partial incoming transcription chunks are displayed live through `TalkText`, while finalized chunks are sent to the back end for processing (see Section A.1.2). GPT-4o is used for: determining semantic chunking of incoming transcripts, `TalkNote`-related processing, and `TalkTip` prompts. Gemini 2.5 Pro is used for visually linking `TalkNotes` to scene elements.

A.1.2 ***Semantic Chunking and `TalkNote` Processing.*** Transcribed speech is segmented into `TalkNotes` through a semantic chunking pipeline powered by GPT-4o. Transcript fragments are buffered, and new fragments are evaluated in context to determine whether they continue the current topic or begin a new topic. Splits are avoided for filler words, minor topic shifts, or short clarifications, and triggered when a new idea, problem, or task is introduced.

When a split is detected—or when the user pauses for more than eight seconds—the buffered text is promoted to a new `TalkNote`. Each `TalkNote` is then processed asynchronously with individual LLM prompts: it is summarized, assigned to one or more process labels, augmented with action suggestions, and checked for potential merging with the preceding note if the user resumes the same line of thought. In parallel, dynamic clustering routines group related `TalkNotes` into `TalkThreads`, enabling users to revisit connected reasoning across time.

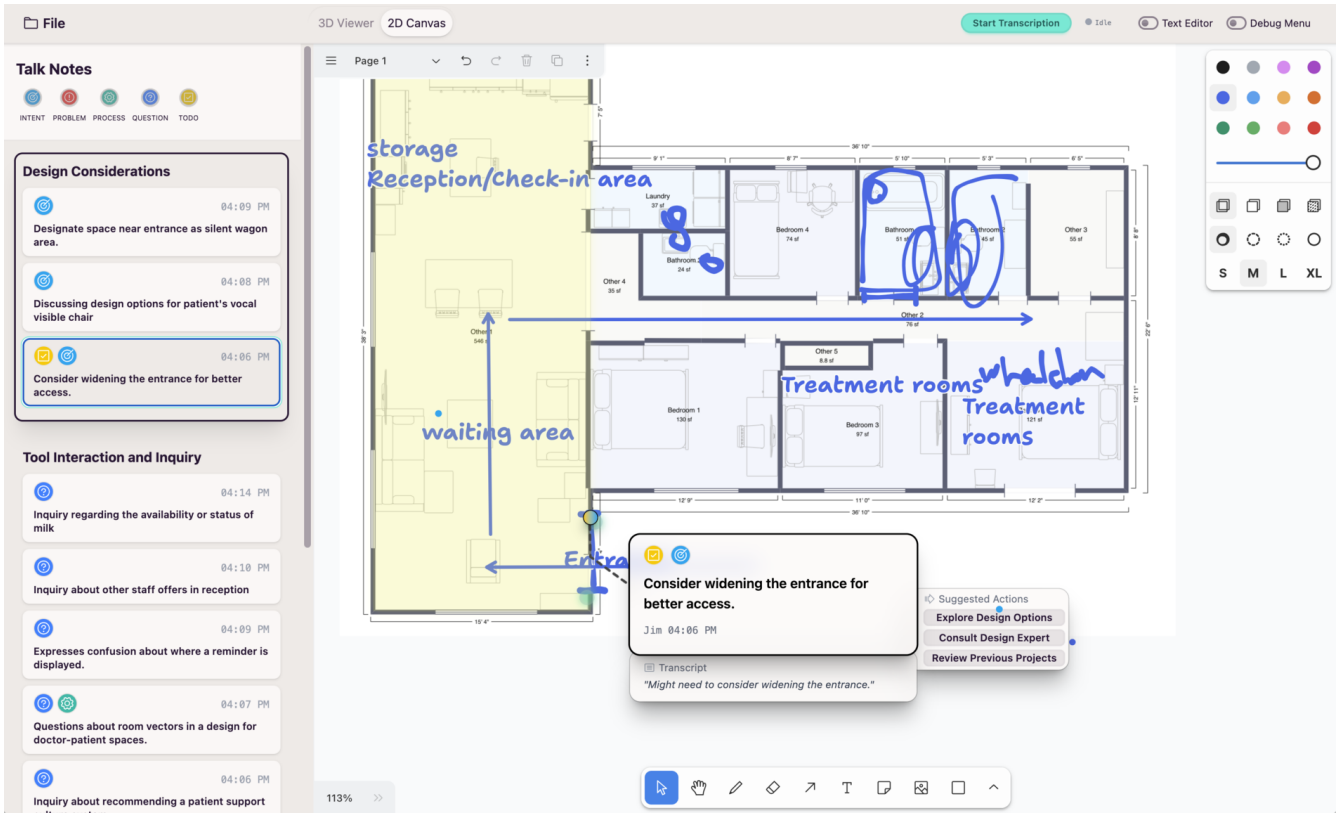
A.1.3 ***Multimodal Context Capture***. Beyond transcription, PointAloud captures contextual signals from the design interaction. Cursor traces are logged and attached to the TalkNote as temporal-spatial metadata. For every TalkNote, the system reconstructs a visual overlay by rendering the relevant canvas screenshot and superimposing circular markers at the recorded cursor locations corresponding to each spoken fragment. This composite image, along with a textual timeline of utterances, pointer coordinates, and the design scene, is then provided as input to the LLM (Gemini Pro 2.5) for identifying referenced regions in the design scene and highlighting specific floorplan areas. This multimodal capture ensures that TalkNotes can later re-situate users in the original design context by visually highlighting both pointer movement and referenced elements.

A.1.4 **Talk*Tip and* Talk*Note Suggestion Mechanisms***. The system periodically generates TalkTip candidate suggestions in the background by prompting GPT-4o with the current transcript and design brief, requesting concise, actionable tips in three categories: *potential issue*, *new idea*, and *probing question*. These suggestions are stored and, at regular intervals, the system queries GPT-4o whether any tip is sufficiently relevant to interrupt the user.

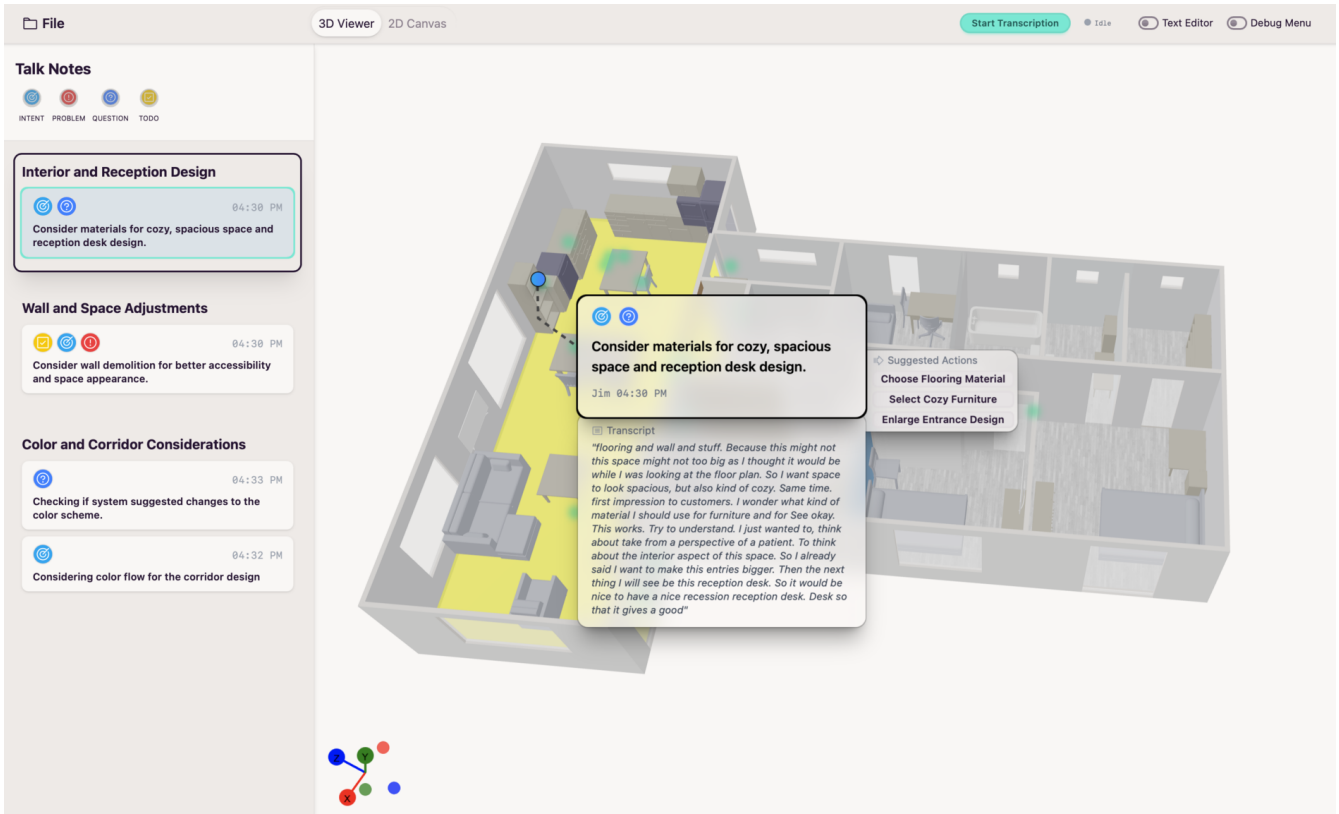
To support resurfacing contextually similar TalkNotes, the system uses the current transcript, previous TalkNotes, and the design brief to query GPT-4o to return the IDs of previously created TalkNotes that are highly relevant to the user's present focus. These notes are then visually highlighted in the interface, enabling users to quickly revisit and build upon earlier reasoning or decisions.

A.2 PointAloud Annotation Task Outcome

This section contains screenshots of the PointAloud system taken at the end of each participant's annotation task from our user study (see study phases 2 and 3 in Section 5.2).

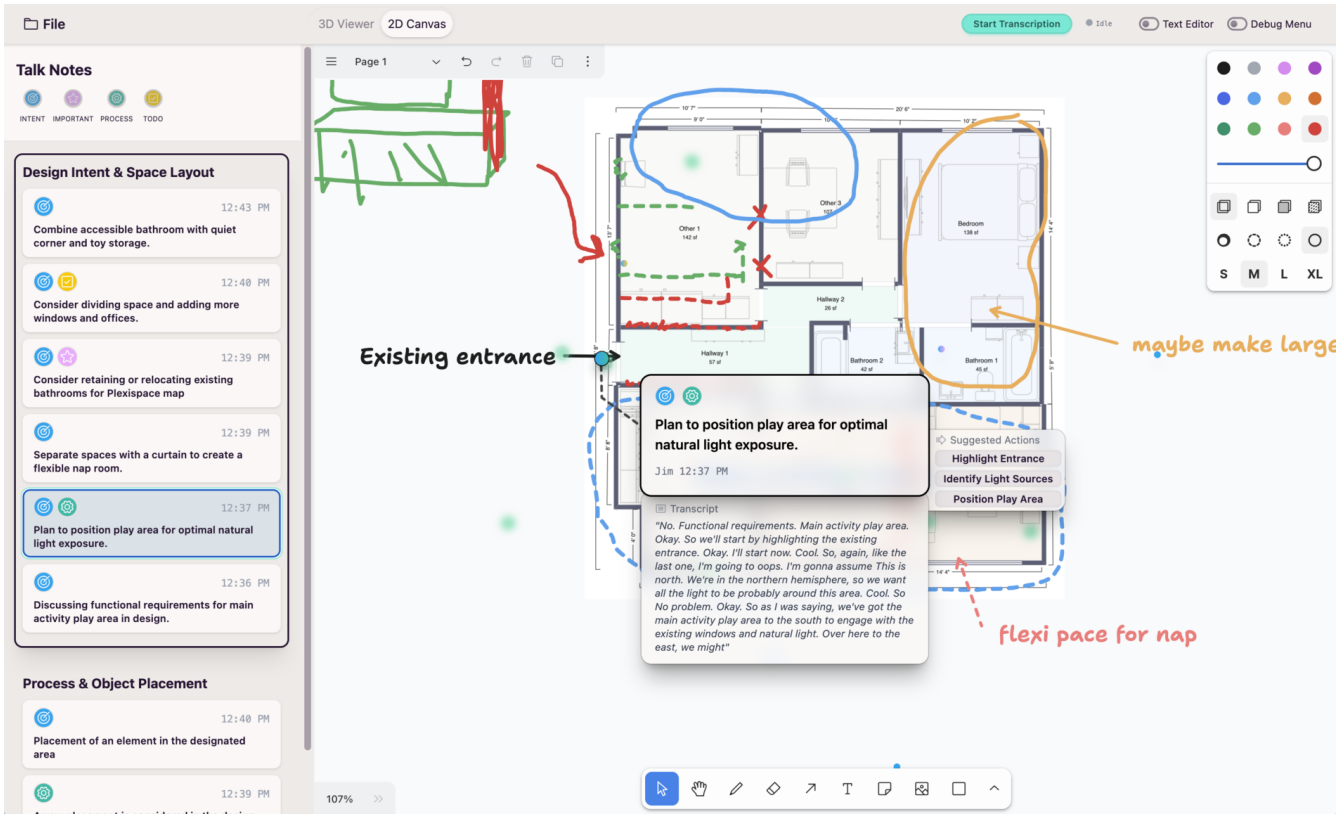


(a) P01 – 2D Annotation Task

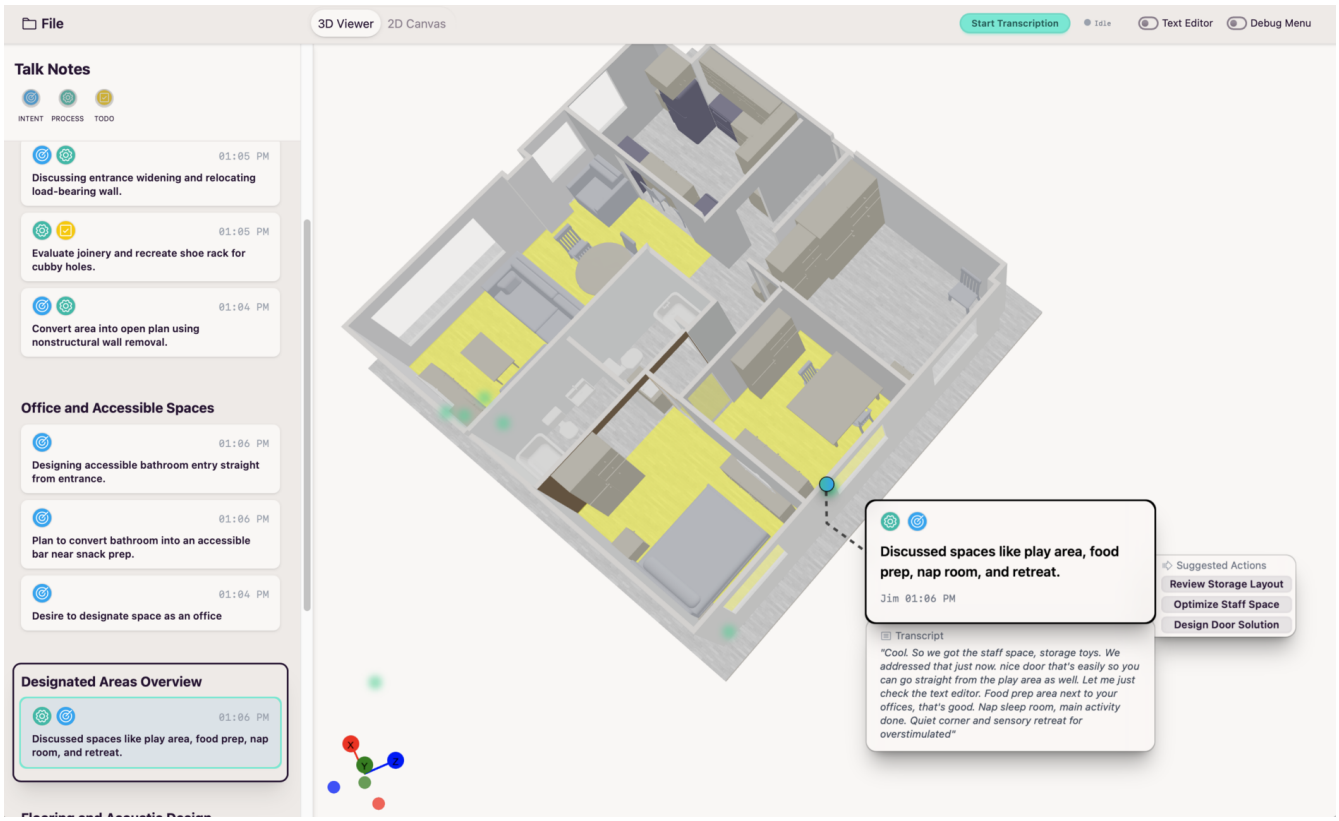


(b) P01 – 3D Review Task

Figure 21: PointAloud annotation task outcomes P01.

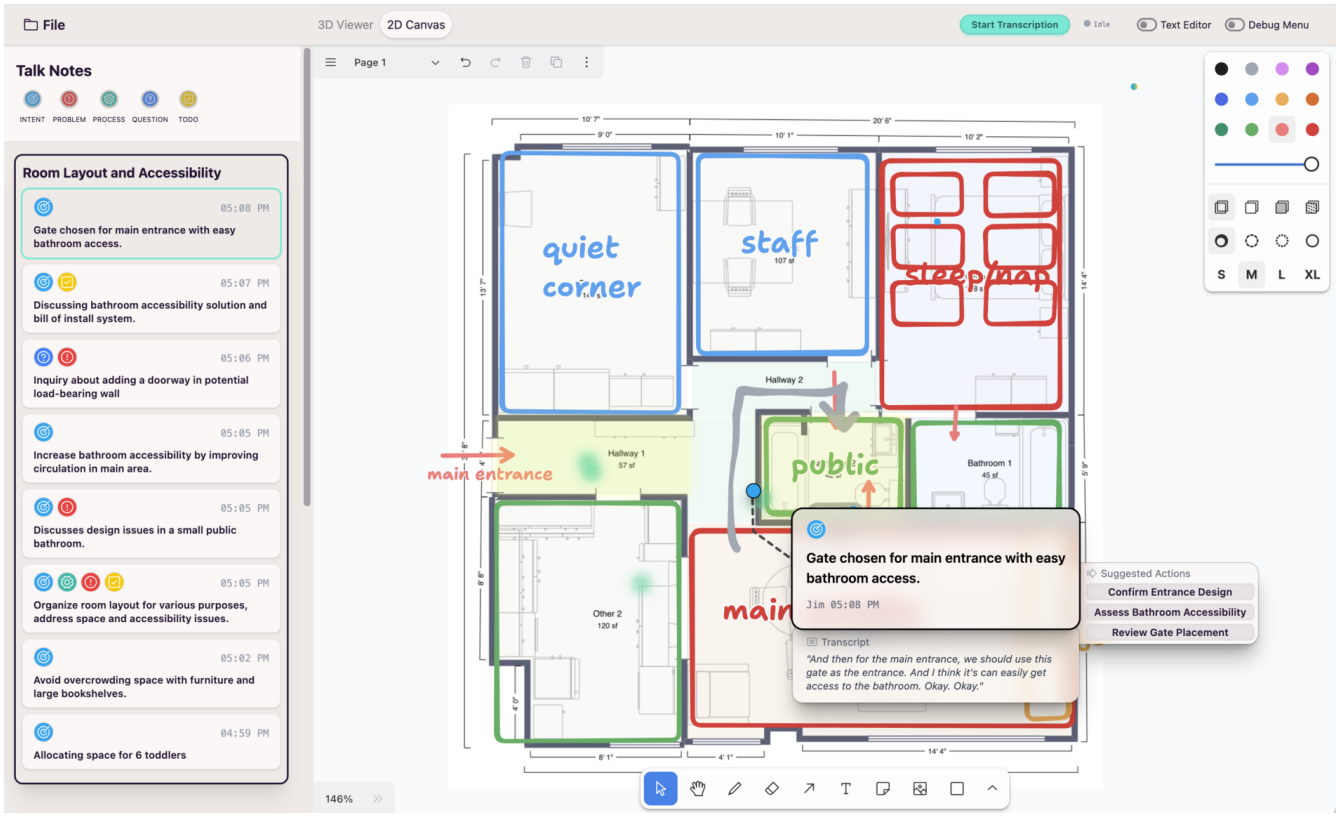


(a) P02 – 2D Annotation Task

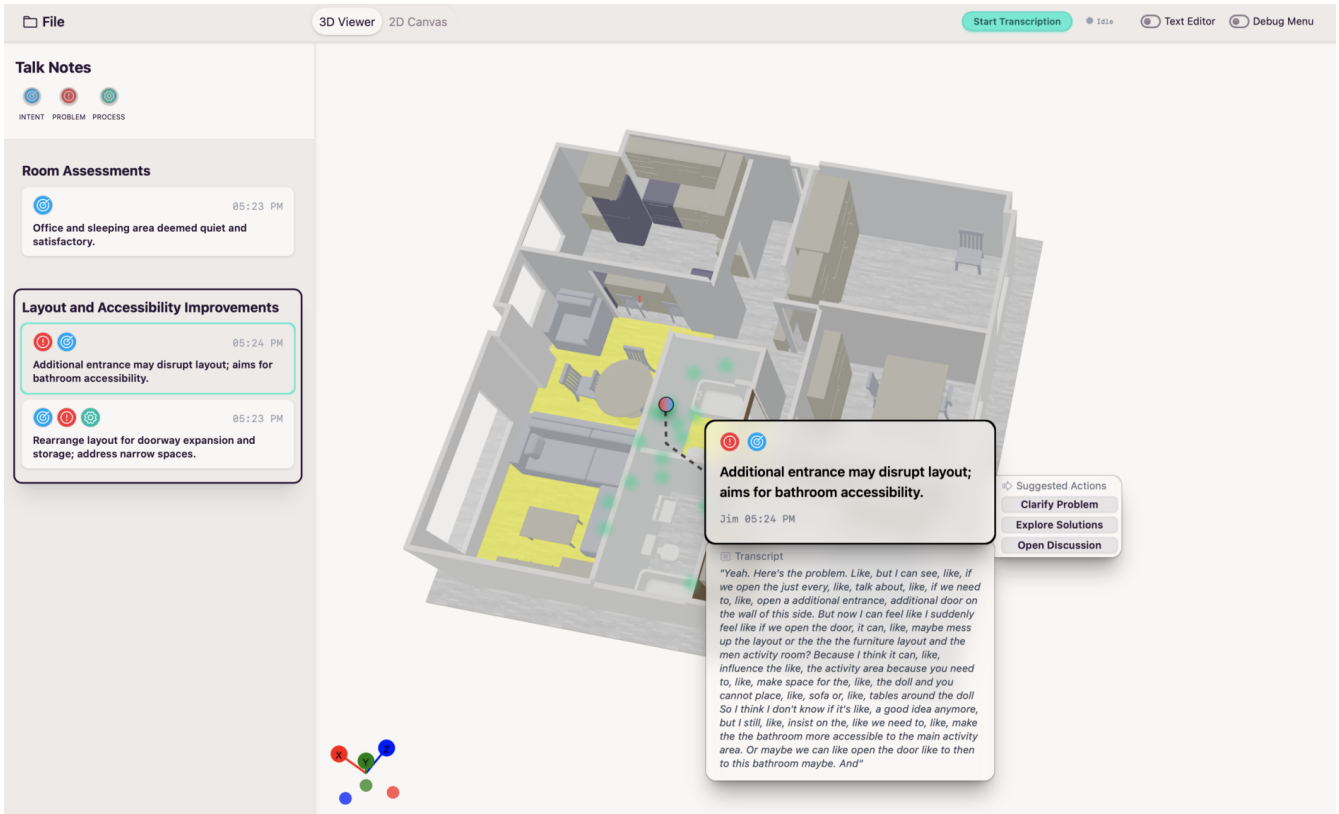


(b) P02 – 3D Review Task

Figure 22: PointAloud annotation task outcomes P02.

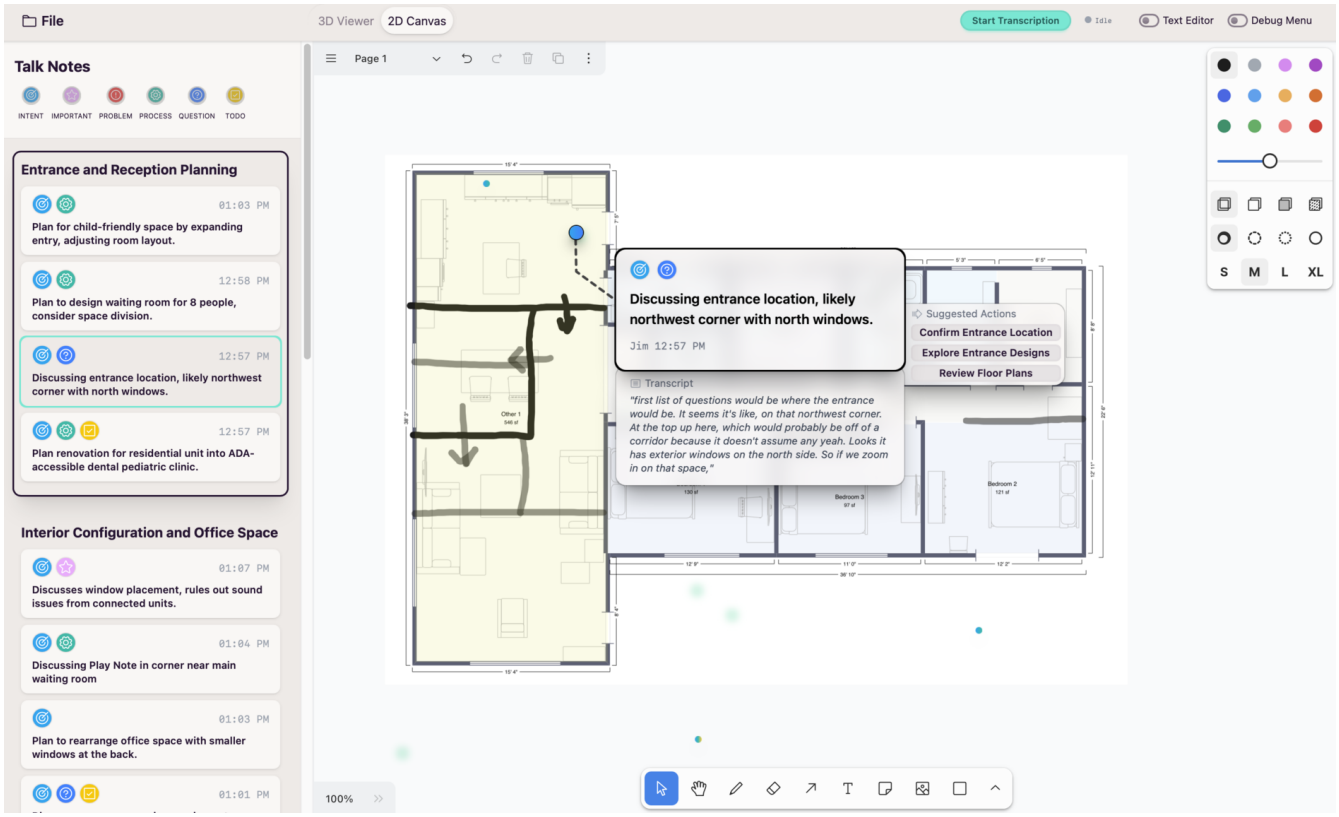


(a) P03 – 2D Annotation Task

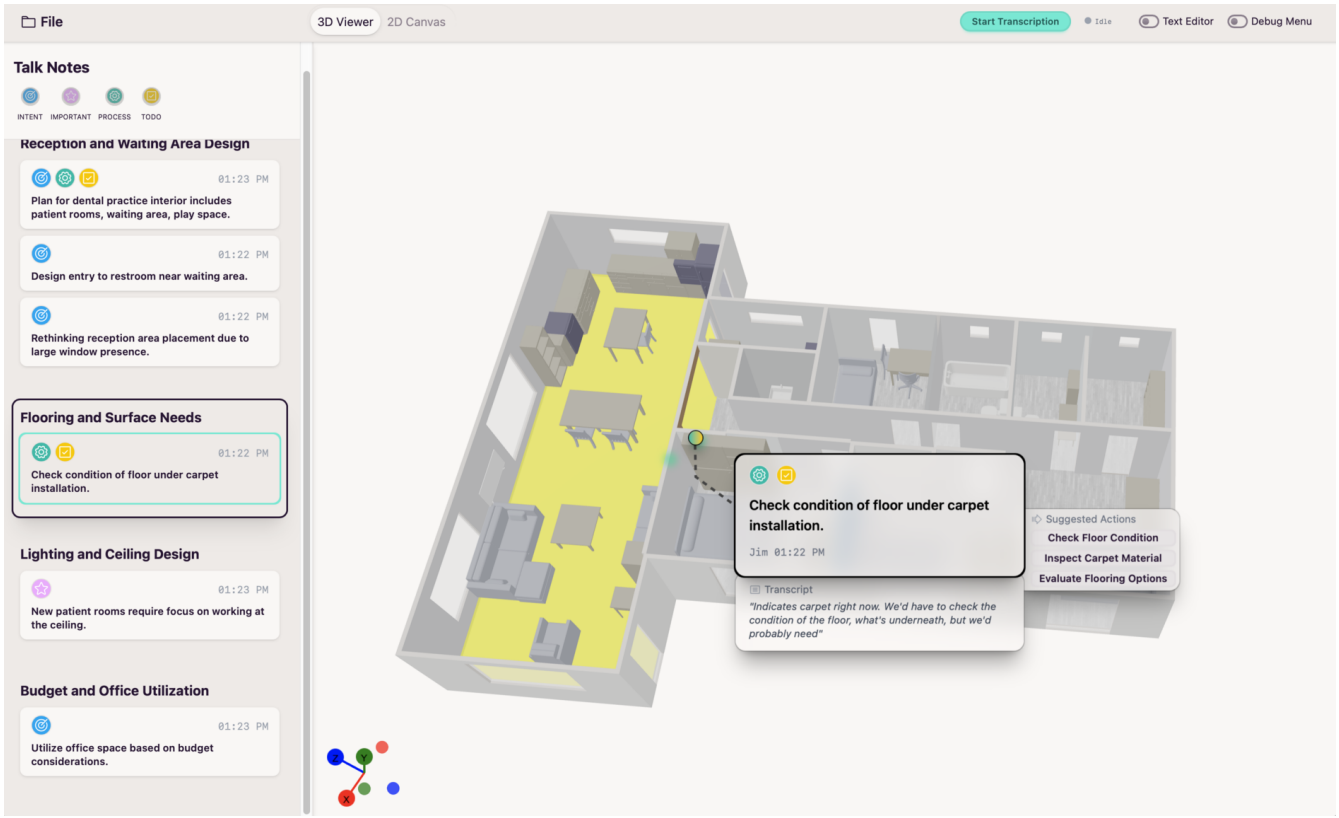


(b) P03 – 3D Review Task

Figure 23: PointAloud annotation task outcomes P03.

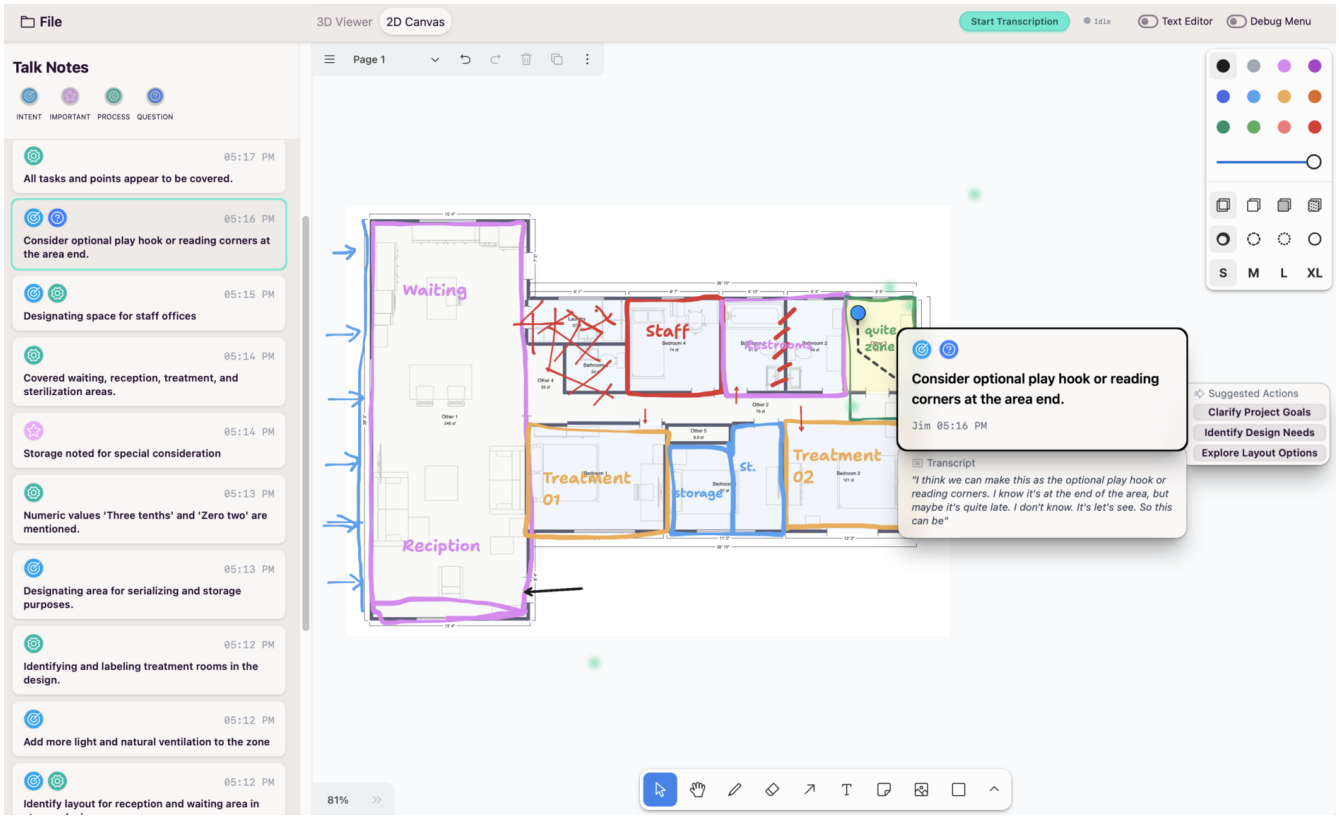


(a) P04 – 2D Annotation Task

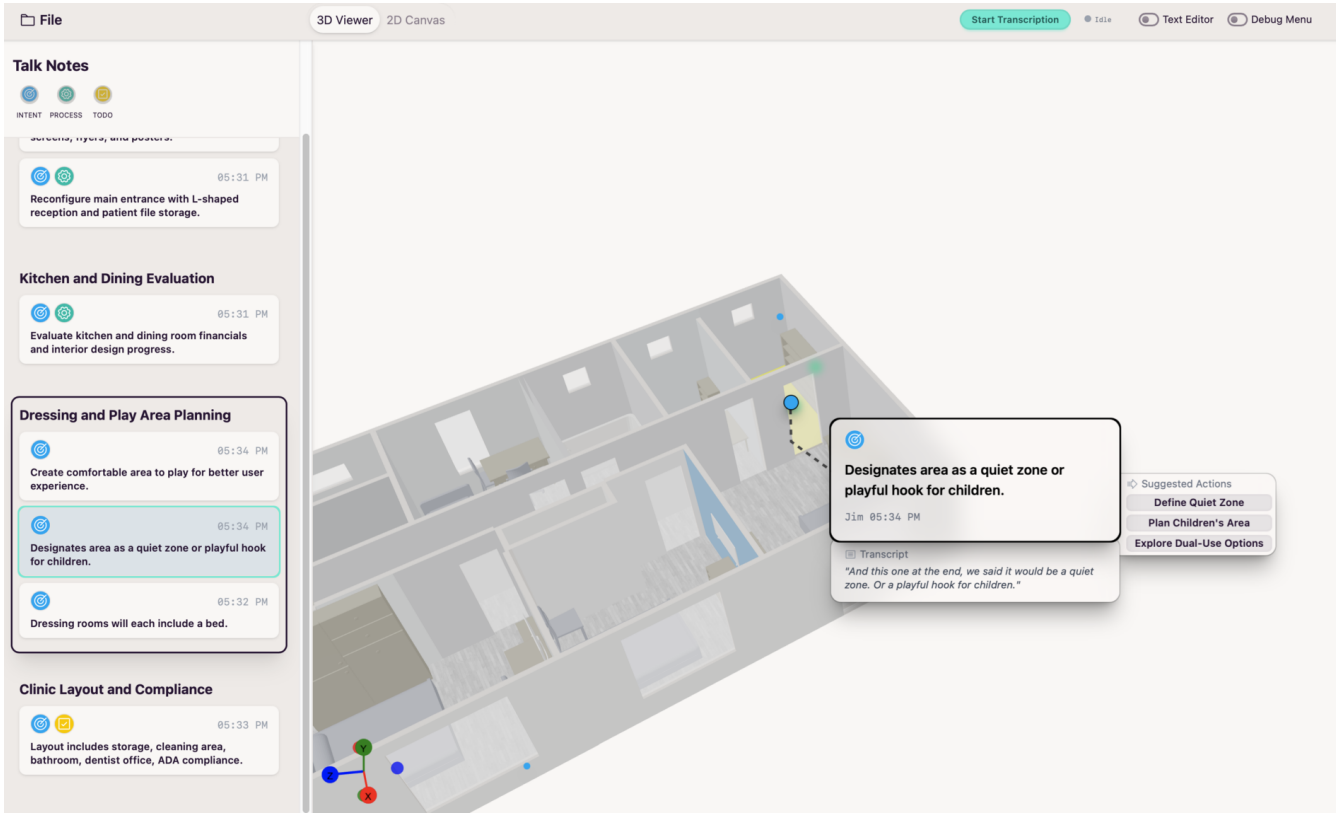


(b) P04 – 3D Review Task

Figure 24: PointAloud annotation task outcomes P04.

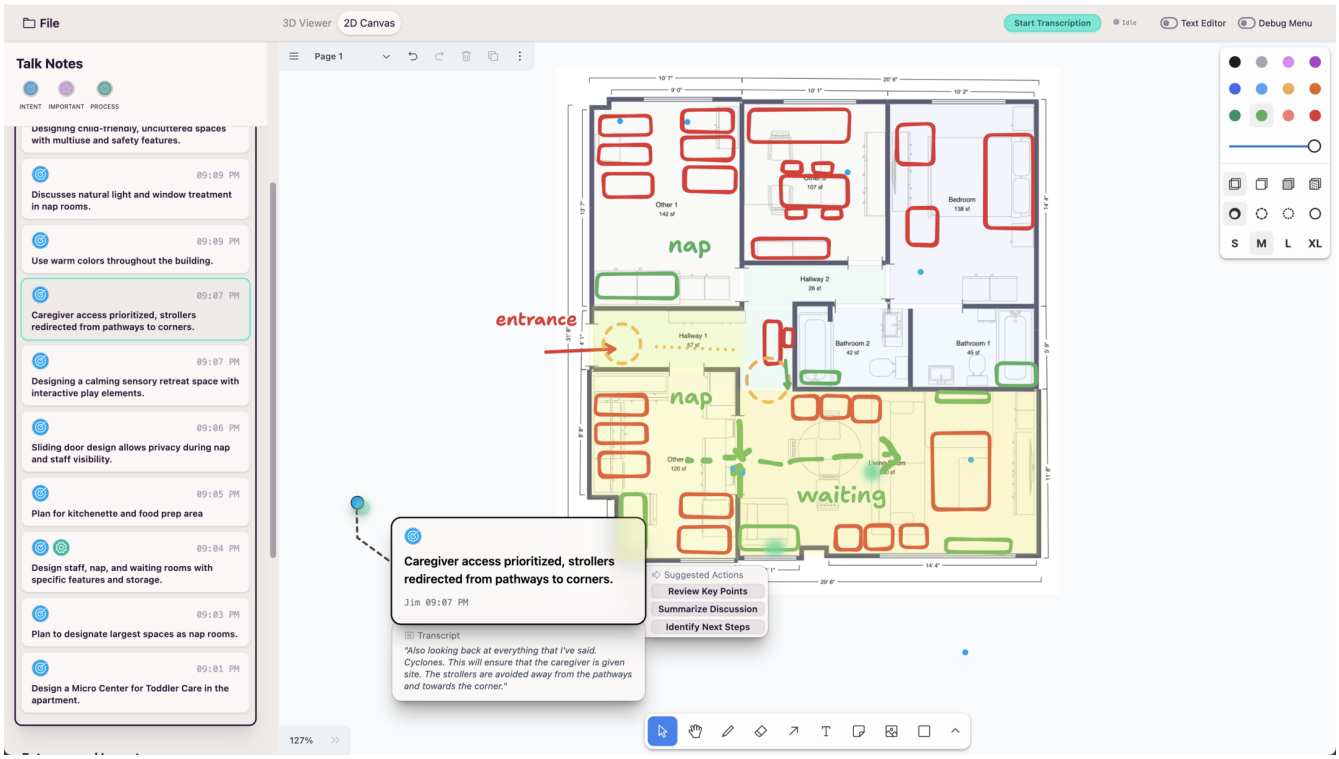


(a) P05 – 2D Annotation Task

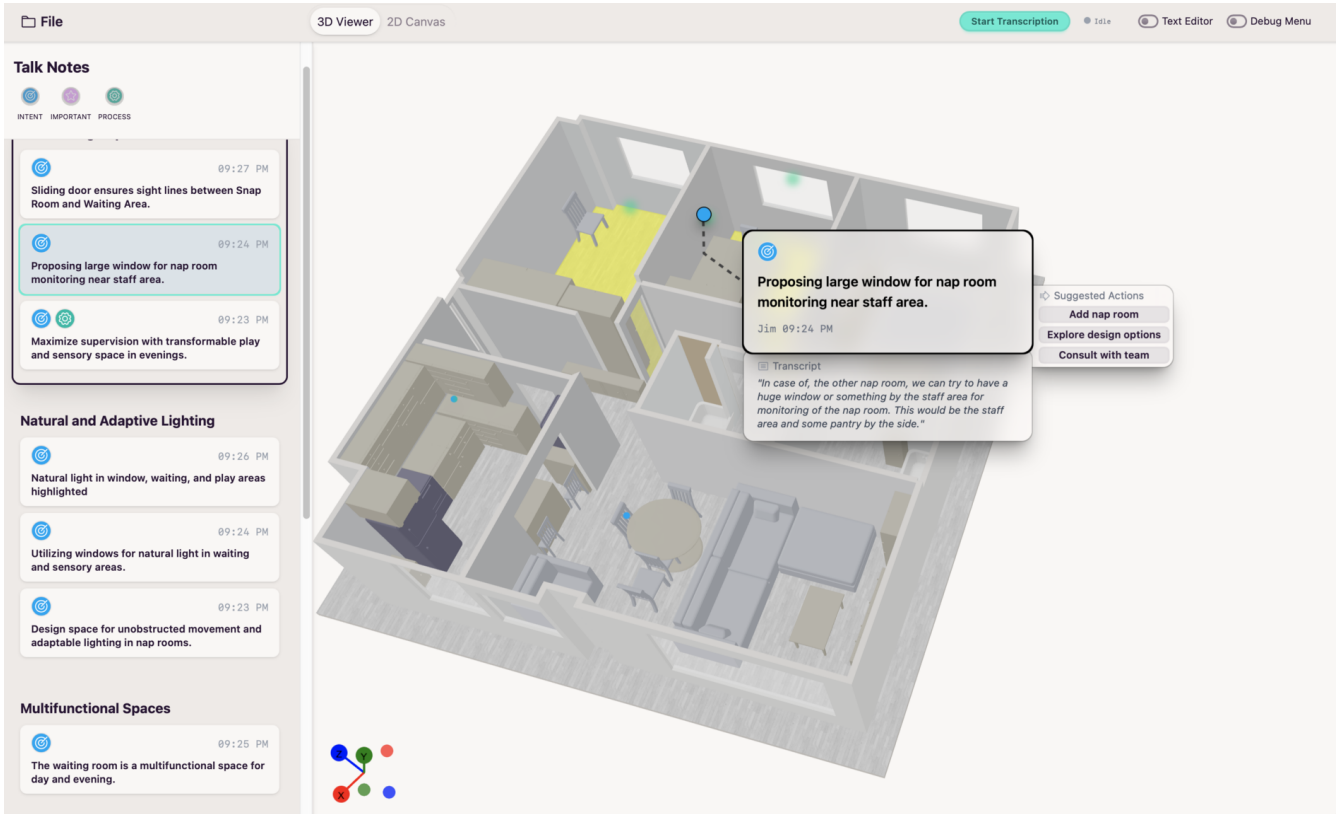


(b) P05 – 3D Review Task

Figure 25: PointAloud annotation task outcomes P05.

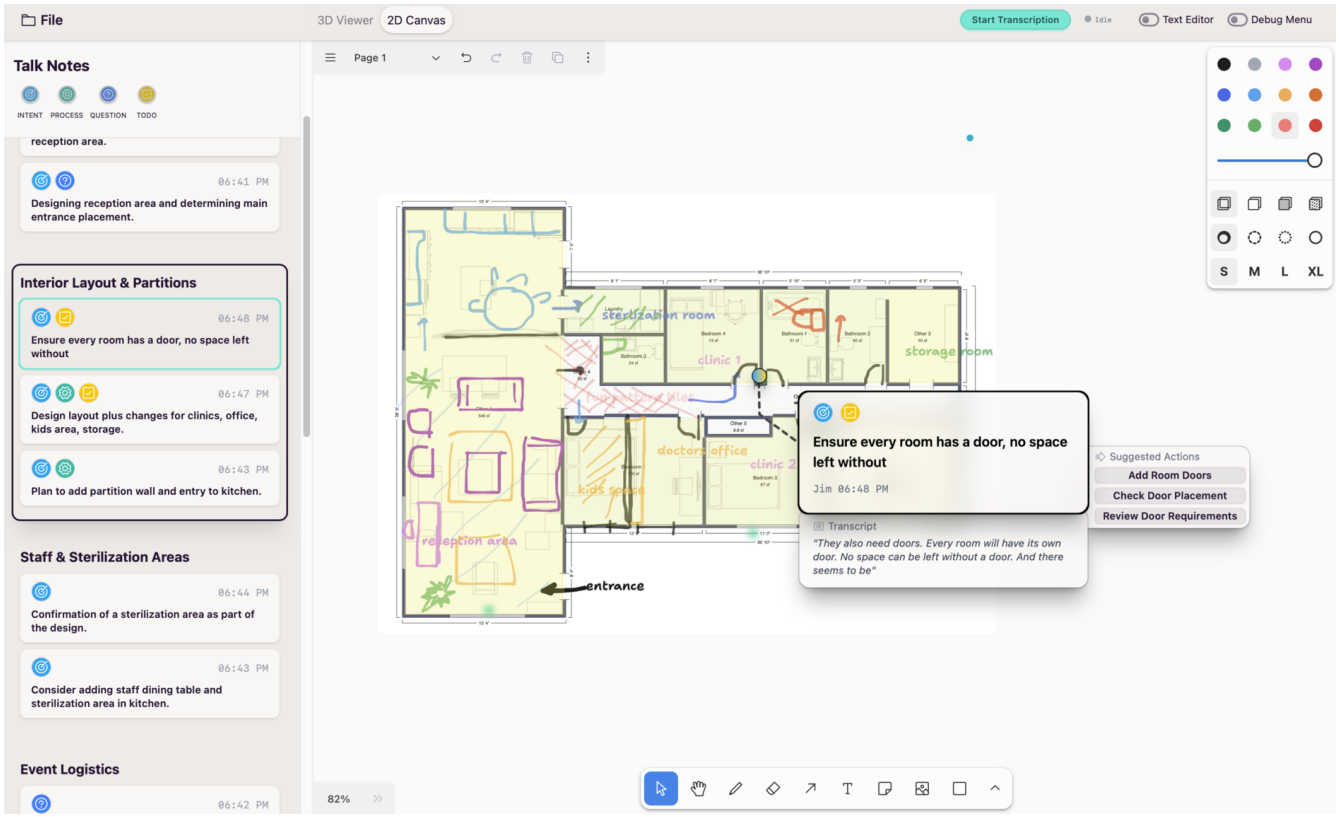


(a) P06 – 2D Annotation Task

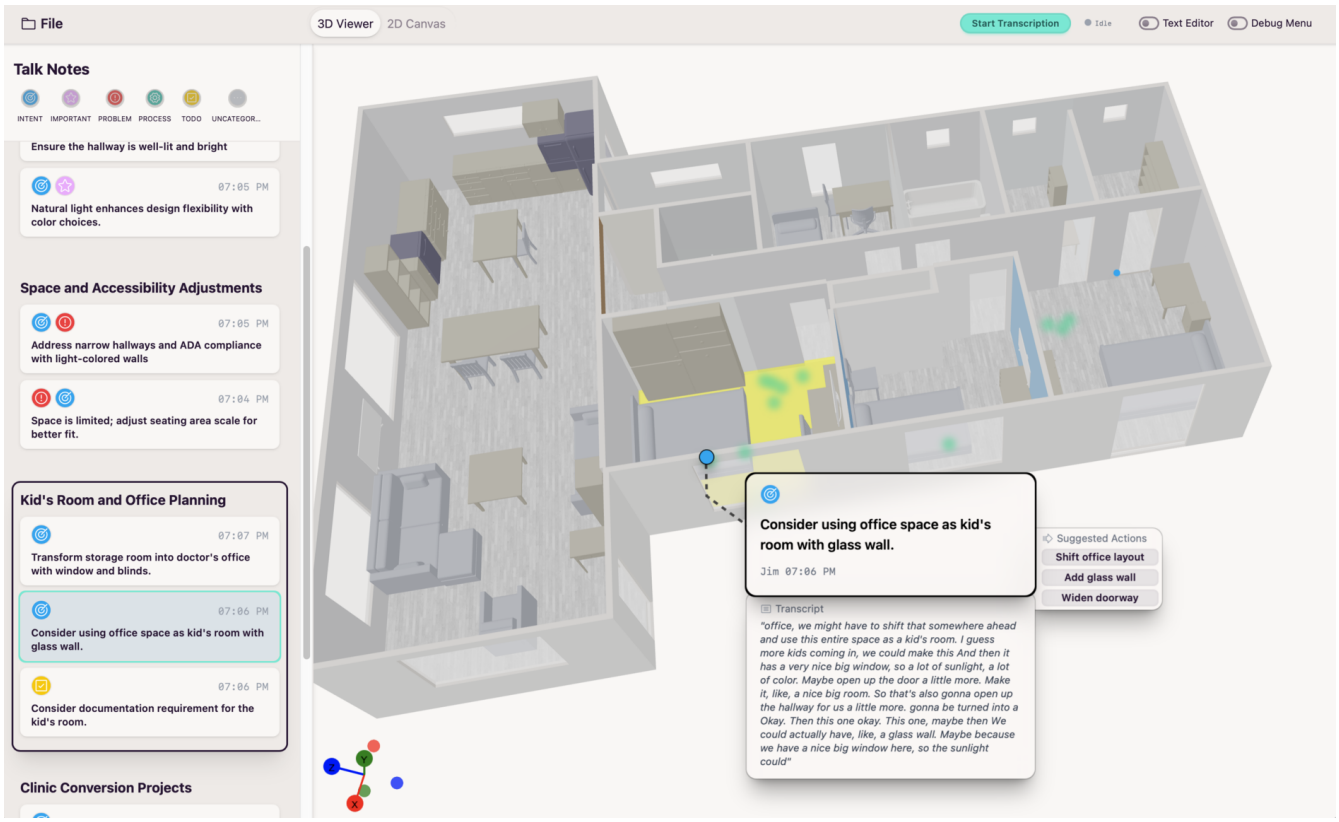


(b) P06 – 3D Review Task

Figure 26: PointAloud annotation task outcomes P06.

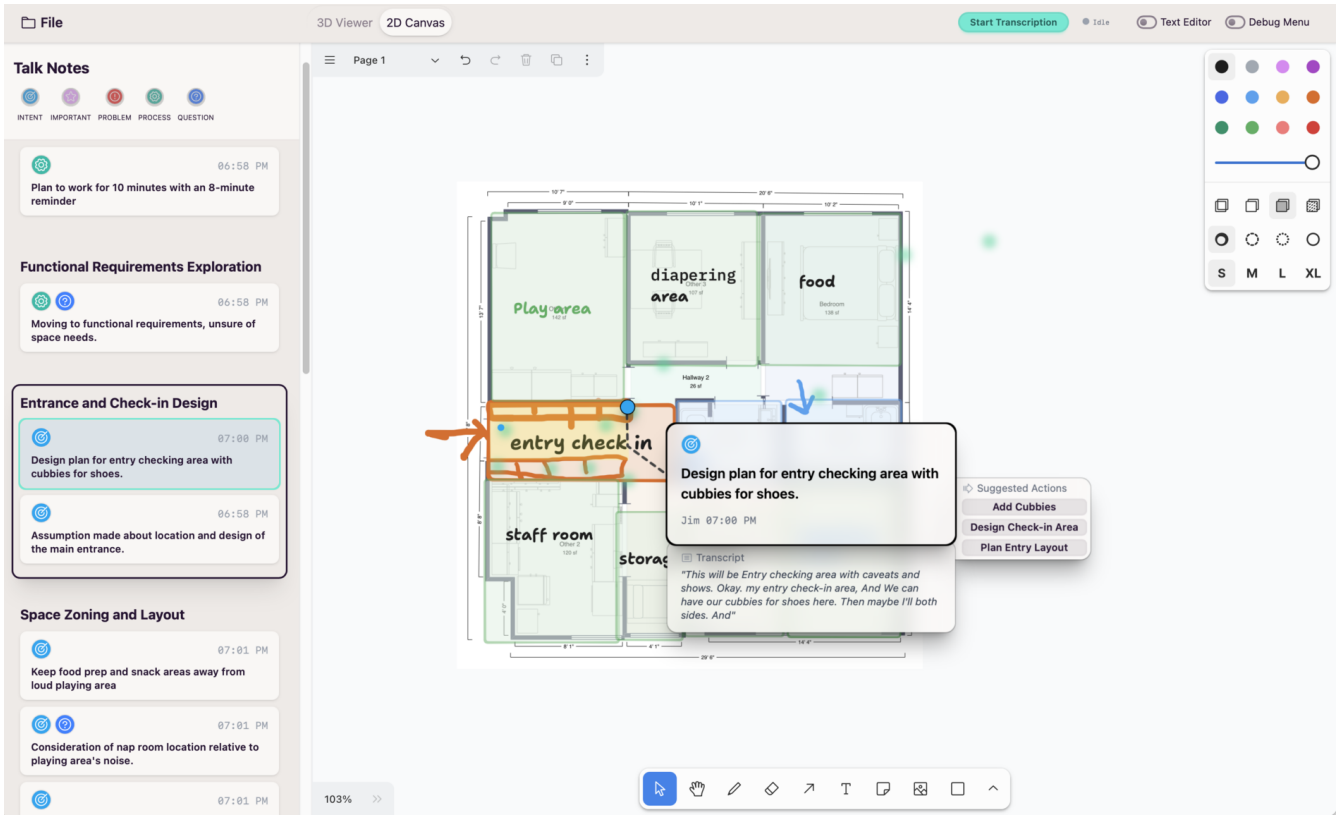


(a) P07 – 2D Annotation Task

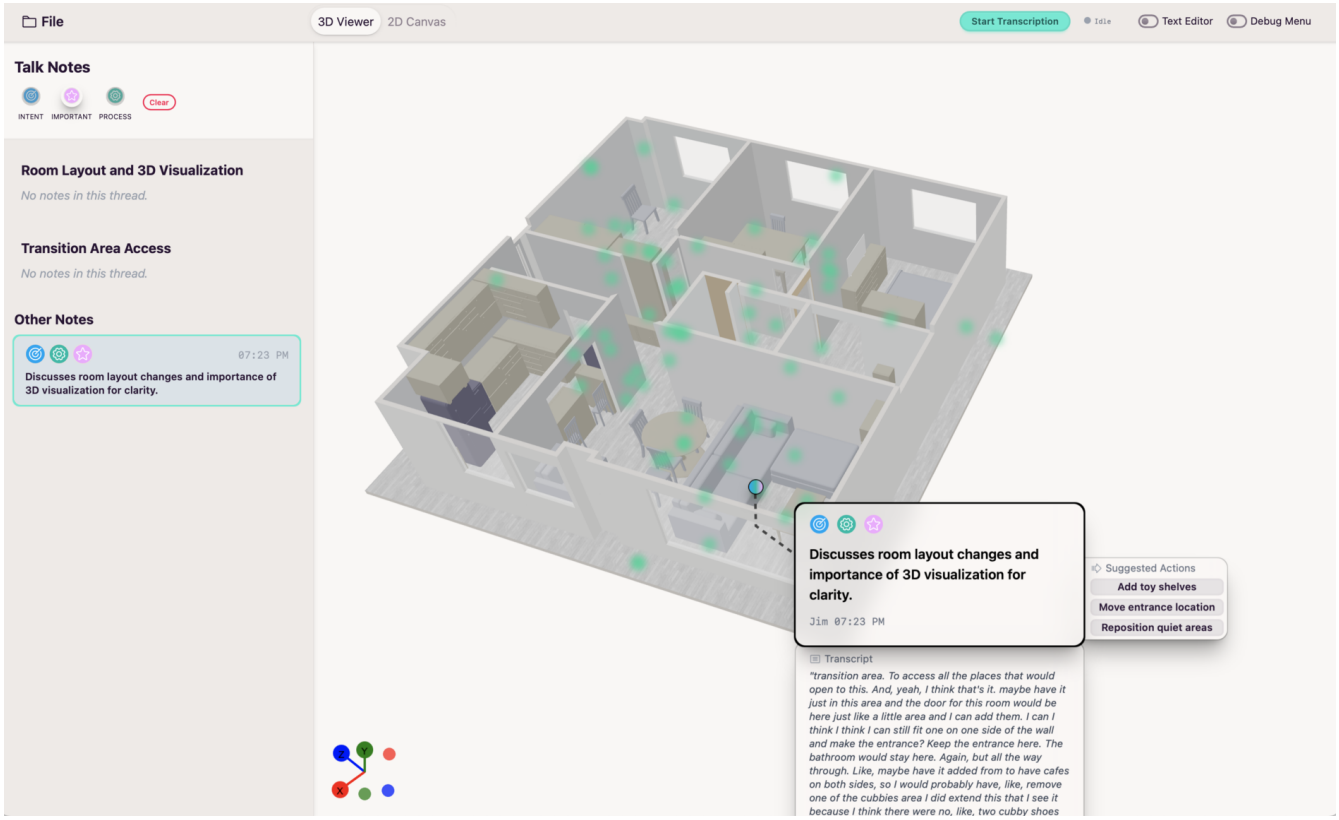


(b) P07 – 3D Review Task

Figure 27: PointAloud annotation task outcomes P07.

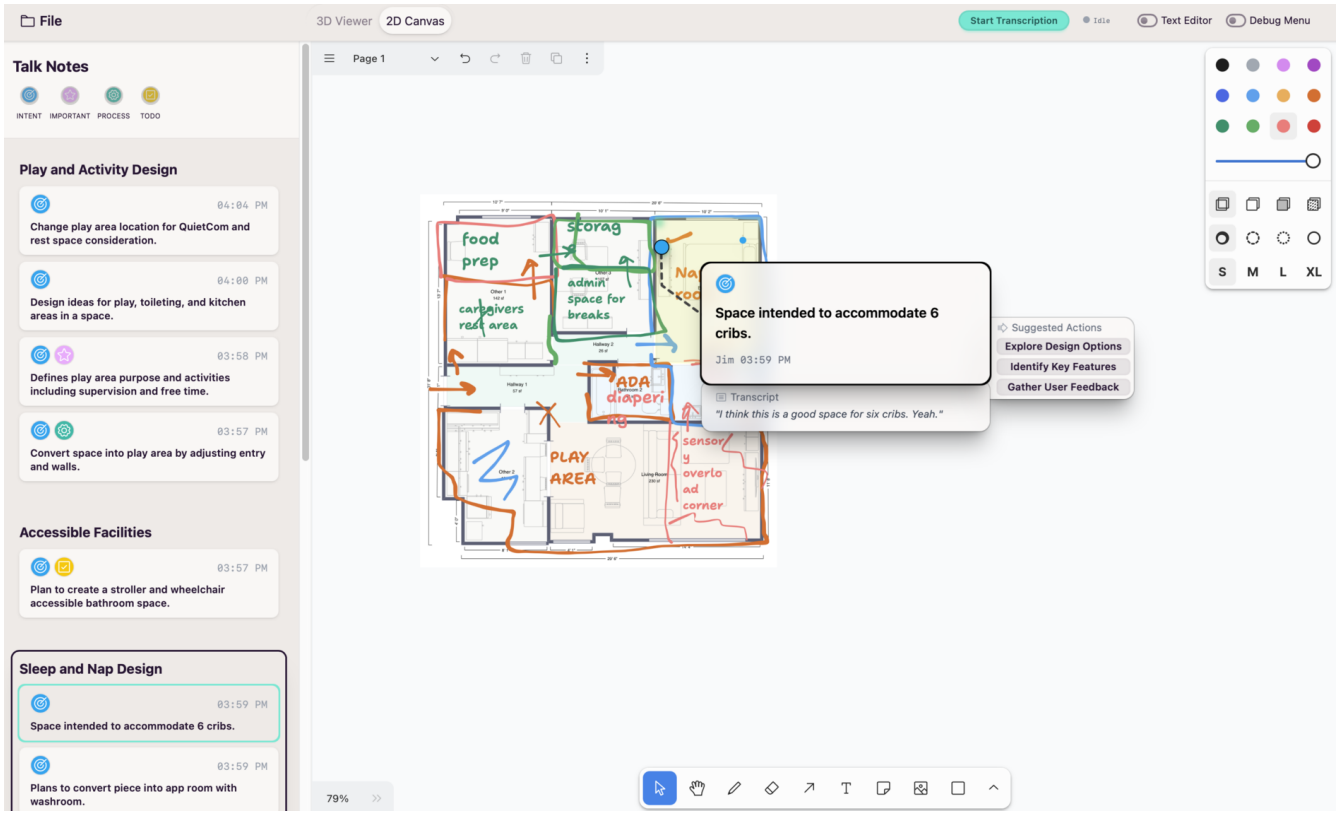


(a) P08 – 2D Annotation Task

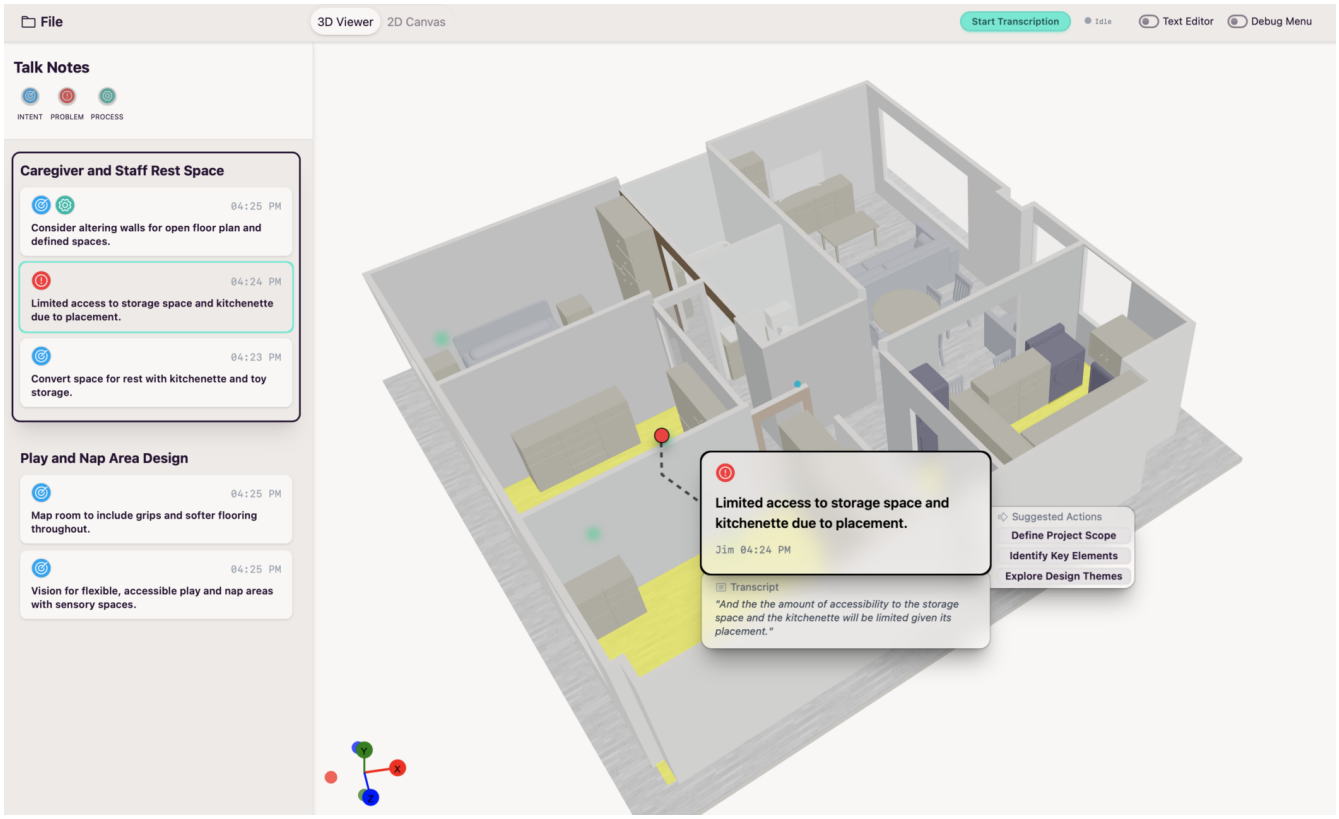


(b) P08 – 3D Review Task

Figure 28: PointAloud annotation task outcomes P08.

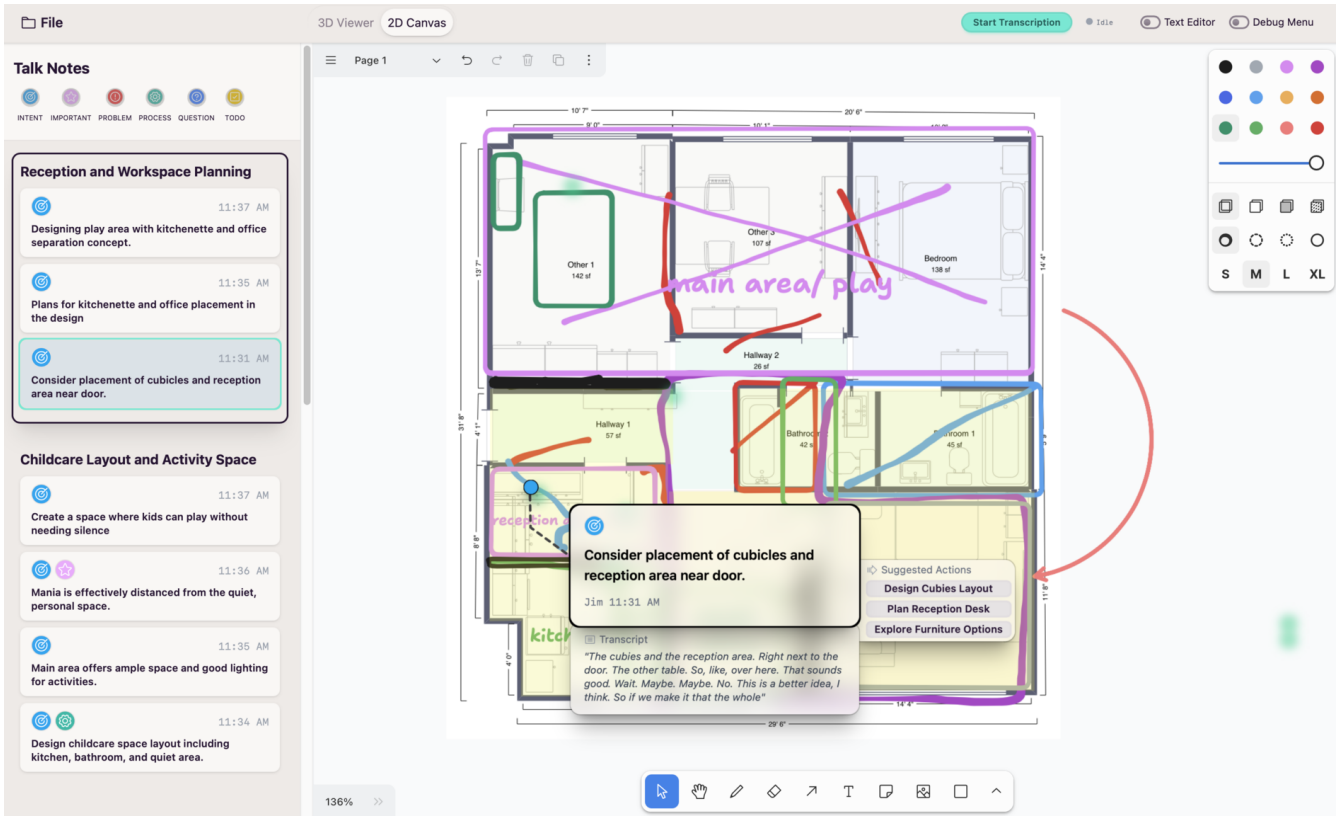


(a) P09 – 2D Annotation Task

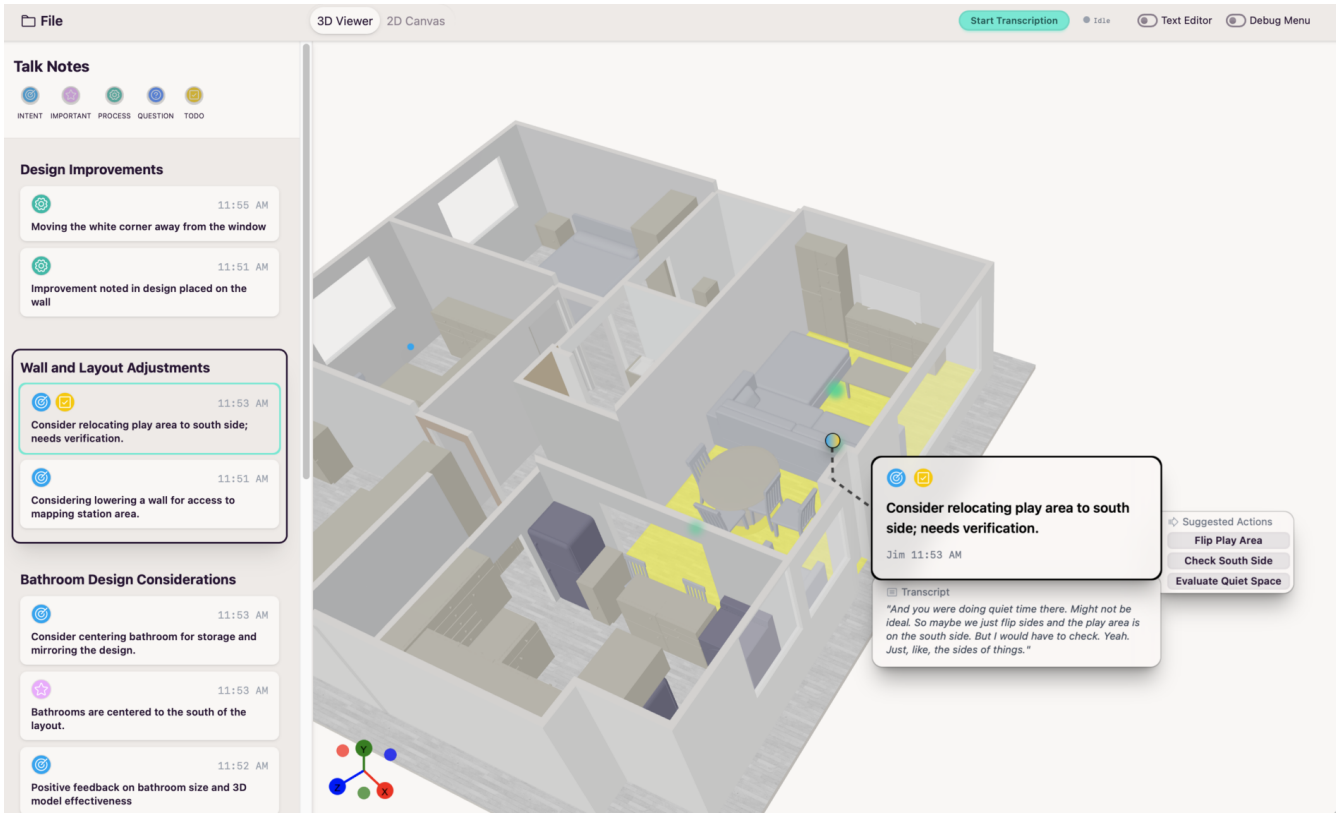


(b) P09 – 3D Review Task

Figure 29: PointAloud annotation task outcomes P09.

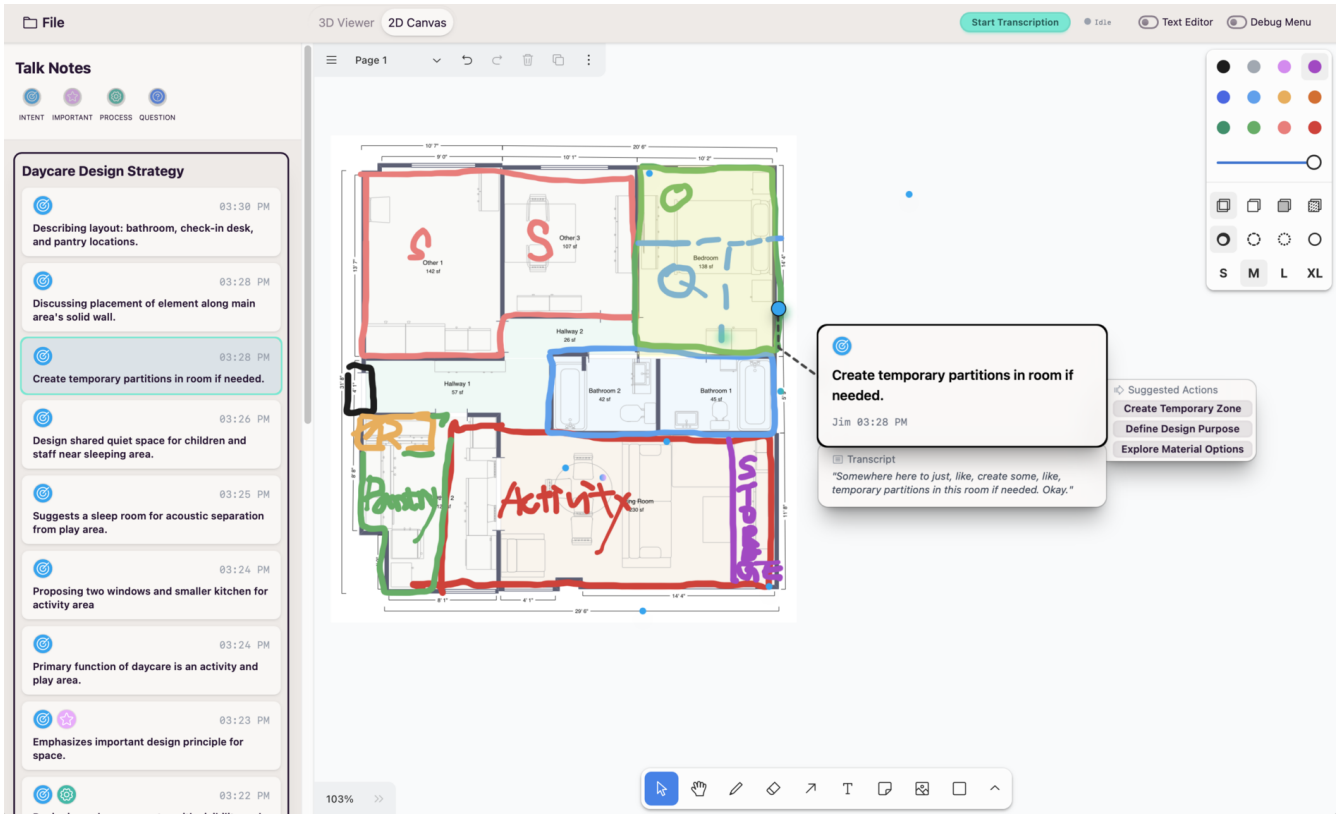


(a) P10 – 2D Annotation Task

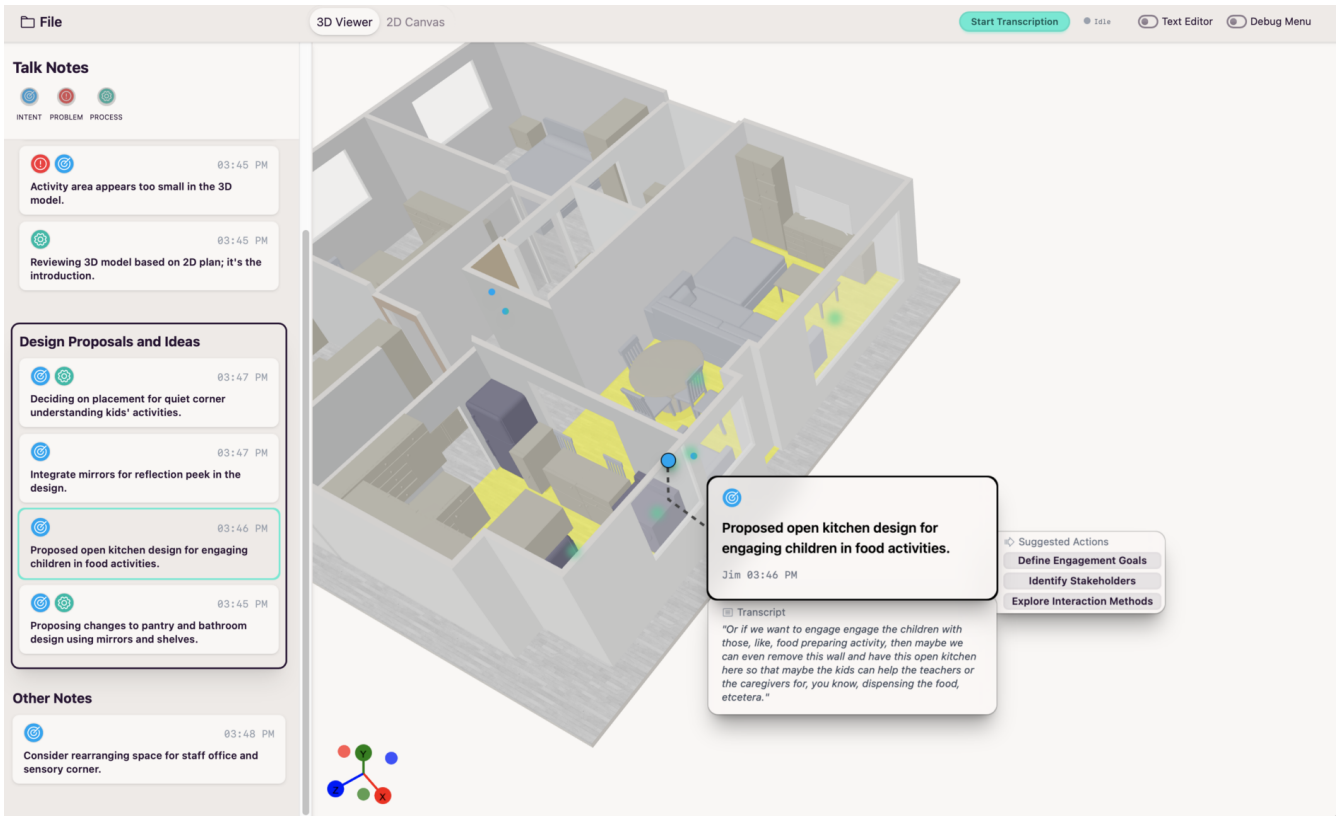


(b) P10 – 3D Review Task

Figure 30: PointAloud annotation task outcomes P10.

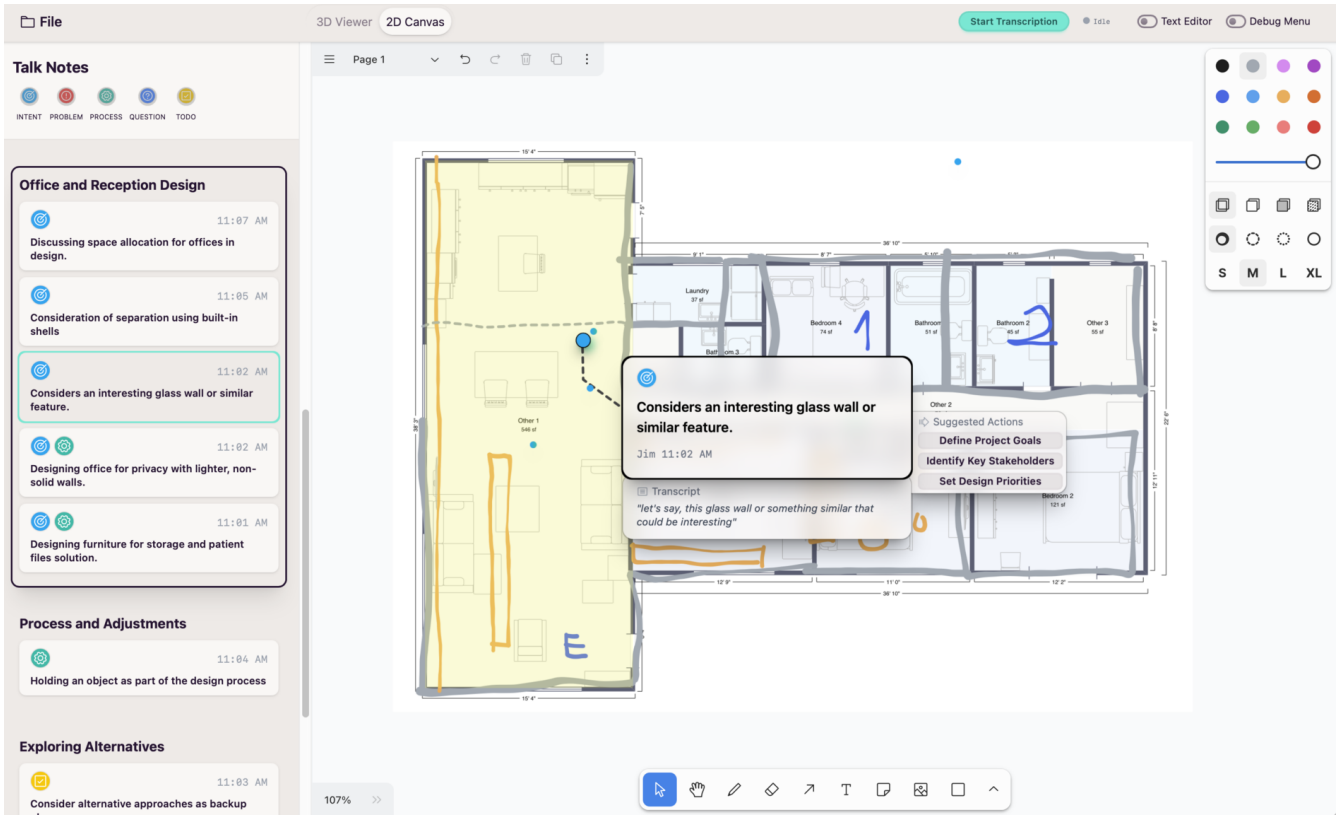


(a) P11 – 2D Annotation Task

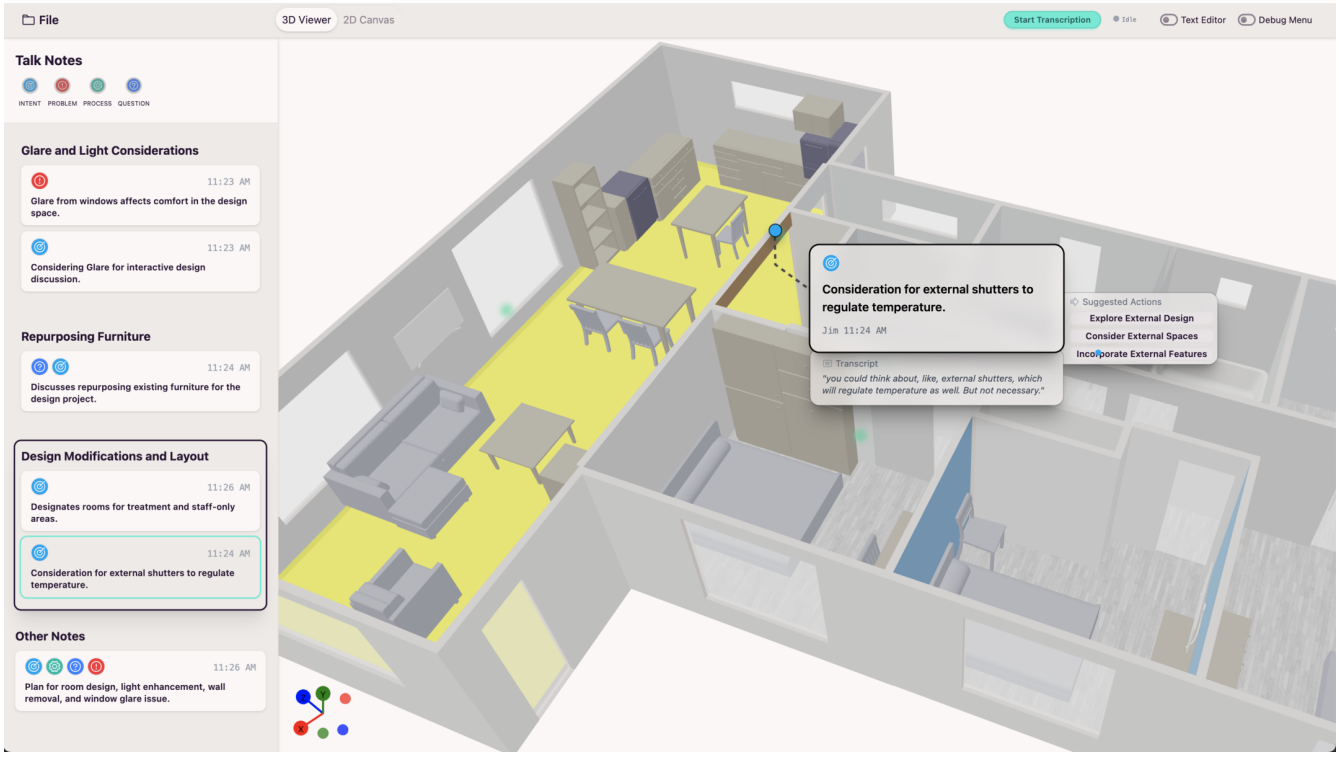


(b) P11 – 3D Review Task

Figure 31: PointAloud annotation task outcomes P11.



(a) P12 – 2D Annotation Task



(b) P12 – 3D Review Task

Figure 32: PointAloud annotation task outcomes P12.