Exploring Real-Time Collaboration in Crowd-Powered Systems Through a UI Design Tool

SANG WON LEE, Computer Science & Eng., University of Michigan
REBECCA KROSNICK, Computer Science & Eng., University of Michigan
SUN YOUNG PARK, Stamps School of Art & Design and School of Information, University of Michigan
BRANDON KEELEAN, Stamps School of Art & Design, University of Michigan
SACH VAIDYA, Computer Science & Eng., University of Michigan
STEPHANIE D. O'KEEFE, Computer Science & Eng., University of Michigan
WALTER S. LASECKI, Computer Science & Eng. and School of Information, University of Michigan

Real-time collaboration between a requester and crowd workers expands the scope of tasks that crowdsourcing can be used for by letting requesters and crowd workers interactively create various artifacts (e.g., a sketch prototype, writing, or program code). In such systems, it is increasingly common to allow requesters to verbally describe their requests, receive responses from workers, and provide immediate and continuous feedback to enhance the overall outcome of the two groups' real-time collaboration. This work is motivated by the lack of a deep understanding of the challenges that end users of such systems face in their communication with workers and the need of design implications that can address such challenges for other similar systems. In this paper, we investigate how requesters verbally communicate and collaborate with crowd workers to solve a complex task. Using a crowd-powered UI design tool, we conducted a qualitative user study to explore how requesters with varying expertise communicate and collaborate with crowd workers. Our work also identifies the unique challenges that collaborative crowdsourcing systems pose: potential expertise differences between requesters and crowd workers, the asymmetry of two-way communication (e.g., speech versus text), and the shared artifact's concurrent modification by two disparate groups. Finally, we make design recommendations that can inform the design of future real-time collaboration processes in crowdsourcing systems.

CCS Concepts: • **Human-centered computing → Collaborative interaction**; **User Interface Design**; **Collaborative content creation**;

Additional Key Words and Phrases: Crowdsourcing; Human Computation; Collaborative Design; Communication; Prototyping; User Interface Design;

ACM Reference format:
Sang Won Lee, Rebecca Krosnick, Sun Young Park, Brandon Keelean, Sach Vaidya, Stephanie D. O'Keefe, and Walter S. Lasecki. 2018. Exploring Real-Time Collaboration in Crowd-Powered Systems Through a UI Design Tool. *Proc. ACM Hum.-Comput. Interact.* 2, CSCW, Article 104 (November 2018), 23 pages.
https://doi.org/10.1145/3274373

1 INTRODUCTION

In recent years, interactive crowdsourcing systems have been designed that allow end users to send requests directly to crowd workers, and to continuously collaborate with them in real time [32, 35].

2573-0142/2018/11-ART104 $15.00
https://doi.org/10.1145/3274373

In such systems, requesters can verbally communicate with workers to request various tasks—ranging from accessibility assistance to writing software—and researchers have developed a number of crowd-powered systems to enable this novel collaborative model [3, 7, 8, 38, 40, 45, 49]. This form of direct and natural interaction between a requester and crowd workers in real time minimizes the limitations of microtask crowdsourcing workflows, where a task needs to be broken down into context-free microtasks and the responses aggregated afterward [46]. Real-time collaboration between requesters and workers is useful in expanding the application of crowdsourcing to broader domains, allowing workers to help solve open-ended tasks whose steps are unknown in advance and may change during the process.

While this mode of end users directly collaborating with crowd workers unlocks new opportunities for crowdsourcing, it introduces new set of challenges for requesters beyond mere unfamiliarity with such collaboration. This is because the characteristics unique to crowdsourcing—ubiquity, scale, transiency, and anonymity—make using crowd-powered systems different from simply collaborating with other users via real-time groupware. Specifically, designing crowdsourcing systems that leverage direct interaction entails an inherently asymmetrical structure between a requester and crowd workers in multiple ways: scale, roles, expertise, communication channels, and user interfaces. Such differences and imbalances are embedded in interactive crowd-powered systems and pose new design challenges beyond those addressed in real-time groupware design [20]. However, the current body of research in real-time crowd-powered systems has focused on developing interactive crowdsourcing systems for specific domains, in which the system that enables workers to resolve certain types of tasks is the main contribution [3, 7, 8, 38, 40, 45, 49]. Furthermore, the validation process of such systems is often focused on the crowd's ability to complete the central task, holding the requester constant throughout each trial in order to control for variance among requesters [11, 38, 45]. As a result, we lack sufficient understanding of the challenges that requesters have as end users of interactive crowdsourcing systems in general. This work focuses on examining the requester side of crowdsourcing.

The goal of this paper is to better understand the challenges that are present when a requester directly collaborates and verbally communicates with crowd workers in real-time collaborative crowdsourcing systems, and to formulate design recommendations from the findings for future interactive crowdsourcing systems. In crowdsourcing systems, requesters are those who specify tasks for crowd workers. Crowd workers then choose and complete the tasks that are available through crowdsourcing platforms, such as Amazon Mechanical Turk [1]. Requesters generally have less control over who will do the tasks, and less direct interaction with the workers, than the traditional recruitment processes. This inherently transient nature of crowdsourcing poses a set of challenges that are not present in an existing real-time groupware. In addition, designing a single system for two disparate groups—requesters and workers—may require different approaches for each group, which can create unforeseen challenges. In particular, we are initially motivated by the following questions: Q1) How do requesters find and understand the benefits of real-time collaboration in crowdsourcing and engage with workers (5.1)? Q2) How do potential asymmetry of expertise, roles, or communication channels between a requester and crowd workers impact their communication and collaboration (5.2,5.3)? Q3) What particular challenges do requesters and workers face in collaboration (5.4,5.5)? Our findings can contribute to improving existing systems and informing the design of future interactive crowdsourcing and mixed-initiative systems (6).

To answer these questions, we conducted a user study investigating how requesters collaborate and communicate with workers via an interactive crowdsourcing system. We chose to use an existing crowd-powered system, Apparition, which allows requesters to create sketch prototypes of graphical user interfaces with crowd workers [38, 45]. In this study, an end user (requester) is asked to verbally describe and draw a GUI sketch; crowd workers behind the scene listen to the description and view the sketch, and create a refined GUI in real time. While the authors of previous works focused on building and validating systems to enable crowd workers to effectively create an interactive graphical user interface prototype [38, 45], in this study we focus on how various requesters use the system and collaborate with workers.

Our core contributions are valuable beyond the scope of crowd-powered design tools, derived from a qualitative analysis using questionnaires, interviews, and observations. The system and task in this study involve common components of real-time collaboration in crowdsourcing: a requester, crowd workers, an artifact (sketch), visual context (canvas) that is shared remotely, and communication channels (voice/chat). This structure is found in many crowdsourcing systems that involve real-time collaboration between a requester and crowd workers, allowing us to generalize our design recommendations. A summary of our findings is as follows:

- Requesters actively collaborate with workers and understand the benefits of working with them in real time (5.1).
- Expert requesters may speak in a style that makes it difficult for crowd workers to comprehend their requests (i.e., be less descriptive, use more jargon) (5.2).
- The asymmetry in communication modalities (speech versus text) causes confusion (5.3).
- Sharing the visual context was not enough to effectively coordinate collaboration (5.4).
- Speaking rates vary among individuals; rapid speech may result in a backlog of requests (5.5).

The design recommendations drawn from the study contribute to the broader goal of enhancing the design of crowdsourcing systems that can facilitate effective real-time communication between requesters and crowd workers.

2 RELATED WORK

Crowdsourcing has been effective in resolving complex tasks that computers alone cannot automate. The most established approach to crowdsourcing involves breaking the task down into microtasks that non-expert crowd workers can solve independently [46]. This method is effective in settings where the problem solving process is known ahead of time and immediate feedback is not necessary, but has limitations in addressing "complex" tasks, where tasks cannot be as clearly and rigidly defined [33]. One way researchers have addressed this problem is by using real-time, continuous crowdsourcing to enable on-going interaction between a requester and crowd workers [35, 45].

2.1 Real-Time Collaboration in Crowdsourcing

Real-time crowdsourcing allows requesters to seek help on demand and get assistance over a much shorter timescale (seconds or minutes) [5, 39]. Researchers have explored tools and methods that make various tasks more convenient. One method to recruit crowd workers quickly is a retainer model, in which crowd workers are recruited in advance and routed when a task is available [5, 16]. VizWiz also demonstrated that a crowd could answer questions to assist the visually impaired in under one minute [7]. Chorus, an intelligent conversation assistant, leverages crowd workers' efforts by having them answer any question a requester asks in an instant messenger [27, 41]. EURECA leverages online crowds who collaborate to help robots robustly identify 3D point cloud segments corresponding to user-referenced objects in near real-time [17]. Bolt uses continuous real-time crowdsourcing guided by a AI agent's understanding of the world to enable super-human response speeds (milliseconds) to address even broader classes of interactive tasks [47].

Real-time collaboration between a requester and crowd workers broadens the applications of crowdsourcing, bringing crowdsourcing into the realm of interactive systems. For example, Legion enables interactive crowdsourcing, coordinating and aggregating crowd workers' responses in real time, and allows workers to resolve tasks that require continuous interaction (e.g., controlling graphical user interfaces in real time) [39]. Continuously interacting with crowd workers is effective when the target task is open-ended and a requester needs to explore various ideas and iterate on them. Such an approach has been used in creating early-stage sketches of graphical user interfaces and making them demonstrate interactive behaviors [38, 45]. Similarly, researchers have proposed an augmented whiteboard that allows a co-located team and remote crowd workers to work together in a brainstorming session [3]. Interactive crowdsourcing systems support end users by permitting natural interaction with the system via speech, and then crowd workers behind the scenes can solve complex requests. For example, some real-time crowdsourcing systems allow end users to leave requests for crowd workers in the form of voice recordings [7, 8, 40, 49], or to directly and continuously communicate with workers via live audio streaming [3, 38, 45].

In practice, crowd workers typically work on a particular task for a short amount of time temporarily and can choose to quit a task and transit to another one frequently. This transient and temporary nature of crowdsourcing makes challenges of supporting real-time collaboration go beyond those addressed in real-time groupware design [20]. For example, it has been shown that mitigating the transient nature significantly can enhance the overall outcome of crowdsourcing: 1) co-located teamwork among crowd workers enhances their collaborative experience when compared to remote, independent crowdsourcing [26], and 2) familiar crowd teams doubled the performance of ad-hoc (unfamiliar) teams [55]. However, much of the existing research works have focused on the feasibility of enabling real-time collaboration between a user and workers for domain-specific applications, overlooking the potential challenges of requesters working with crowd workers. Our goal in this work, given the challenging nature of crowdsourcing, is to better understand how a requester communicates and collaborates with crowd workers in the context of creation tasks through various real-time communication channels.

2.2 Supporting Communication and Collaboration in Crowdsourcing

Facilitating communication between a requester and crowd workers is one way to improve the outcome of crowdsourcing. For example, the literature shows that giving workers feedback on a task outcome encouraged them to revise their solutions and yielded a better final outcome [11]. Similarly, Bigham et al. showed that requesters preferred continuous interaction with crowd workers [7]. Chorus:View demonstrates the same benefit of continuous communication with significant speed-up in the completion time [40]. Some researchers developed computational systems that allow requesters to form a team of expert crowd workers to dynamically manage highly complex tasks such as animation creation [54, 61]. Similarly, the Huddler system assembles a stable team of familiar crowd workers to facilitate collaboration on complex tasks [55]. Salehi et al. further explored various communication mechanisms to study how workers can communicate asynchronously with a requester for continuous writing tasks [56]. Oftentimes, the validation of these systems used a controlled set of requesters for the purpose of ensuring request consistency. This experimental setup asks a single person (often an author) to play the role of the requester, having a fixed, scripted task [38, 45, 54–56]. While the thread of research confirmed that these systems are able to coordinate a crowd's effort, it did not fully reveal the potential challenges in communication and collaboration between requesters and crowd workers, nor does it allow us to measure the variation in how different requesters make requests.

3 APPARITION SYSTEM DESCRIPTION

In order to investigate the challenges requesters face in communicating and collaborating with crowd workers, we conducted a user study using Apparition, a web-based sketch prototyping tool [38, 45]. In prior user studies for validating the systems, one of the authors played the role of the requester to eliminate the variability that having multiple requesters would introduce, while

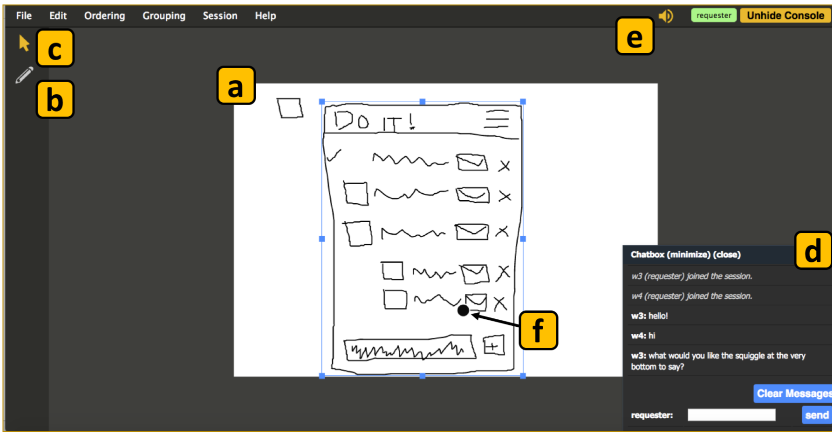

Fig. 1. The requester interface of Apparition supplies a variety of tools that facilitate sketching and communication with crowd workers: (a) shared canvas, (b) pencil tool, (c) selection tool, (d) chat box, (e) mute/unmute button, (f) circles to visualize requester click events. For the worker's interface, see [38, 45].

crowd workers were recruited on-the-fly from Amazon's MTurk platform [1]. While this type of study design is beneficial to confirm if a system is functionally valid, the effectiveness of systems with diverse end users in practice remains unexplored. We focus on the challenges that requesters with varying backgrounds face in using a crowd-powered design tool. In this section, we briefly introduce the features that are designed to support communication and collaboration.

In Apparition, a requester can naturally and continuously interact with crowd workers to create an early-stage graphical user interface prototype. The requester's role is to focus on verbally describing and roughly sketching the desired GUI prototype. The canvas is shared in real time, so crowd workers can see the requester's in-progress sketch and hear their verbal description (Fig. 1-a). State synchronization occurs at a per-element level—a sketched element becomes visible to workers once the requester release the mouse button, completing a segment. Through an interactive tutorial, crowd workers are asked to listen to the verbal description, inspect the rough sketch that the requester draws, and *replace* each sketched element with a higher fidelity shape. Fig. 2 shows an example (T1) of what a requester can draw on a canvas, and demonstrates how crowd workers gradually convert a rough sketch (T2-a) into a more refined one (T2-c).

Apparition provides two different sets of tools for requesters and crowd workers. The requester is able to focus on verbally describing and roughly sketching the desired GUI prototype. They are provided with two tools for sketching: a *pencil tool* (Fig. 1-b), with which the requester can draw free-form drawings; and a *selection tool* (Fig. 1-c), with which the requester can select any element on the canvas to transform (move, resize, rotate) or delete. The tools provided to the requester are kept simple, to minimize the amount of new software that needs to be learned, so that the requester can naturally describe and sketch the desired GUI. In contrast, crowd workers have access to a larger number of functions than requesters do, to enable them to create higher fidelity GUI prototypes. They can add various types of GUI elements (e.g., buttons, arrows, text boxes), and change their properties (color, opacity, stroke width, and stroke pattern). These functions are similar to those available in commodity prototyping tools and slide-based presentation software (e.g., PowerPoint). Therefore, crowd workers with some experience using such software can begin working with these tools immediately. For more details on the worker interface, see [38, 45].

Apparition supports communication between a requester and crowd workers in two distinctive ways. First, it implements real-time audio streaming using WebRTC for a requester to directly speak to crowd workers [4]. The creators of Apparition wanted requesters to have the advantages of multi-modal interaction in simultaneously speaking and sketching to expedite the creation process, similar to collaboration with a co-located colleague [38]. However, audio streaming is one-sided in the case of Apparition, so crowd workers cannot speak to the requester. This design decision was made for Apparition because streaming audio from crowd workers can cause problems: the crowd's voices may drown out the requester's, or the conversation may digress quickly if there are multiple people talking to each other. On another note, a requester can mute their microphone (Fig. 1-e) to stop their audio stream as needed, or to avoid distracting workers. When muted, its icon changes to . The envisioned interaction of the original system resembles that of an intelligent system in which a requester can describe their interface idea, and their idea comes to life in a short span of time [38]. However, in practice, crowd workers often want to communicate with the requester to ask questions, to clarify the request, or to report technical glitches. To that end, Apparition includes a text-based message interface that resembles a typical instant messaging application (Fig. 1-d) [45]. A requester can view messages and respond to them either verbally or in text.

It is worth noting that this kind of asymmetry in communication modalities (text versus speech) is not common in other groupware and social computing contexts. Similar setups can be found in new emerging media such as game streaming services (e.g., Twitch) or other live streaming services (e.g., Facebook Live, Periscope), which recently have been studied for their effects in fostering participation and engaging spectators [22, 23].

Lastly, the shared canvas in Apparition, which is synchronized in real time across requester and crowd worker UIs, not only allows both groups to collaborate on a single, final copy in real time, but also builds conversational grounding among workers and requesters [13]. In addition to the shared canvas, Apparition provides a way to visualize the requester's mouse clicks; when a requester clicks a location on the canvas, it will generate a colored circle at that location which is shown to both the requester and crowd workers. The circle fades out after a few seconds (Fig. 1-f). Apparition allows the requester to point at particular elements or specific spots on the canvas to help in describing GUI requirements to workers.

Apparition takes on the general the structure of other similar systems, particularly a requester and crowd workers collaborate to create a digital artifact in real time. The structure includes: 1) an end user (requester) who leads the communication with concrete goals, 2) a set of crowd workers, and 3) an interactive system for collaboration and communication, the components of which are 3-a) a shared (visual) artifact, 3-b) communication channel (audio streaming and chat), and 3-c) coordination tools that supports fluid collaboration. The findings from the study are applicable to crowdsourcing systems for various applications that have the same structure. In the following section, we describe the method of studying possible challenges in communication.

4 USER STUDY

The goal of this work is to better understand the way in which requesters communicate and collaborate with workers. We conducted a user study with Apparition and ran a qualitative analysis, examining multiple aspects of the communication between requesters and crowd workers to identify the challenges and common patterns that people exhibit during collaboration. Our method draws ideas from observational studies in collaborative design [59, 60]. In particular, the setup of Apparition is similar to the case where distributed workers communicate through a shared visual context [13]. Here, we introduce the detailed study procedure, participants, and our limitations.

4.1 Study Procedure

We conducted the study with 10 participants recruited locally and 20 crowd workers recruited from Amazon Mechanical Turk, an online crowdsourcing platform. The 10 participants came to

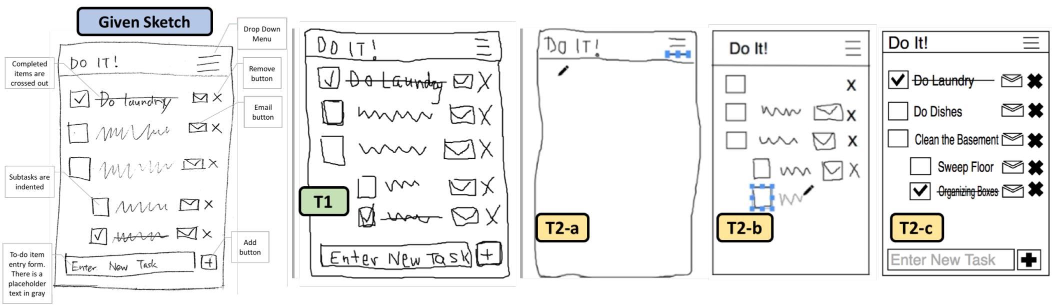

Fig. 2. (**Given Sketch**) The sketch of a to-do list application given to participants (requesters). (**T1**) A sketch hand-drawn in Apparition in Task 1 without crowd workers. (**T2**) Progression of a sketch in Apparition in Task 2: T2-a) the initial hand-drawn by the requester (same drawer as in T1), T2-b) the requester continues hand-drawing and crowd workers begin replacing hand-drawn pieces with higher-fidelity shapes and icons, and T2-c) the crowd workers replace all hand-drawn pieces.

the lab and completed tasks using Apparition as requesters. First, the requesters completed a brief demographics survey and were given instructions on how to use Apparition. Requesters were then asked to complete two tasks. The order in which the tasks were completed was counterbalanced. In both tasks, requesters were shown a low-fidelity prototype (Fig. 2-Given Sketch), which they were asked to describe and sketch, to communicate GUI requirements. We chose the to-do list application primarily for its familiarity with a broad population; therefore, non-expert requesters could quickly understand what they were asked to create. We presented a low-fidelity prototype to the requesters because they were asked to convey the general idea of the UI without needing to meet small details. We included brief annotations in our example GUI in order to avoid participants asking the study moderators clarification questions directly, which may influence the ways how requesters verbally described the GUI.

Here are the descriptions of the two tasks given to the participants during our lab studies:

Task 1 (T1) - Hypothetical Asynchronous Scenario (no Crowd Workers): *Requesters were shown the example GUI and were asked to draw and verbally explain the interface in Apparition as if they were giving a task to someone who would be creating a sketch prototype of the GUI at a later time. Requesters were told the goal of T1 was to convey the sketch idea and to help the hypothetical colleague create a polished prototype of the example GUI.*

Task 2 (T2)- Real-time Collaboration with Crowd Workers: *Requesters were shown the example GUI sketch and were asked to draw and verbally explain the GUI to crowd workers who would be creating a sketch prototype of the GUI in real time via Apparition. Requesters were told the goal of T2 was to create a polished prototype with the help of the crowd workers.*

For both tasks, we asked the participants to stop working on the task when they were satisfied with the outcome; we stopped task execution ourselves only if it took over half an hour. The study moderators then conducted a follow-up interview to gain further insights into the requesters' experiences. We asked requesters questions about their experience communicating and collaborating with crowd workers, as well as their experience using Apparition.

The work primarily focuses on the qualitative analysis of T2, the task in which a requester directly collaborates with crowd workers. Having a hypothetical scenario of a user making a request asynchronously (T1) simply gave us a reference point to better understand the communication styles used in T2, such as how sparse a requester's speech is and how requesters choose to communicate in response to workers making changes in real time. In addition, we wanted participants to experience making a request asynchronously to help them better reflect on the advantages and disadvantages of synchronous work when asked during the follow-up interview. As our primary goal was to analyze potential differences in requester communication style for asynchronous (T1) versus synchronous (T2) collaboration, we were not as interested in how workers' completed requests when they were not working in real-time with the requester; therefore, we did not have crowd workers complete the requests in T1.

4.2 Participants

4.2.1 ***Requesters***. Ten requesters were recruited to complete the user study. Five were designers (denoted by P1, P2, P3, P4, and P5) and five were non-designers (denoted by P6, P7, P8, P9, and P10). Four designers and one non-designer were female. The average age of the designers was 30 (ranging from 27-37), while the average age of the non-designers was 24 (ranging from 19-38). None of the non-designers had previous experience designing a GUI, while all the designers had prior experience in some design-related field, for example: UI, UX, product/industrial, or graphic design. The designers varied greatly in experience, ranging from 1 to 12 years of experience. Each participant was compensated for their time with $15 in cash after the session.

4.2.2 ***Crowd Workers***. We recruited 20 unique crowd workers from Mechanical Turk who had not previously used Apparition, were all U.S.-based, had an approval rating higher than 80%, and had previous experience using slide (or drawing) software. Each session had two crowd workers for the full duration, with the exception of P8's session, during which one of the two workers left mid-way through. Of the 20 workers, 19 completed a demographic survey. Their average age was 32, and 8 of the 19 crowd workers were female (11 male). When rating their design expertise on a five-point Likert scale, the crowd workers' average score was 1.72 ($\sigma = 1.07$), with "1" representing little design experience, and "5" representing experienced professional designer.

The crowd workers were recruited at the beginning of T2, when the moderator of the study posted tasks in MTurk using LegionTools [16, 36]. Workers found to be eligible were asked to watch a tutorial video and were then routed to the task page. Once there were enough crowd workers in a session (i.e., two workers), the moderator checked if workers knew how to use the basic functions of Apparition and trained them by going over a sample task. Once the workers completed the sample task, the participants started the main task (T2). After the task, the workers were asked to fill out a survey and got paid at the rate of $0.17 per minute ($10.20 per hour).

4.2.3 ***Knowledge of Background***. Note that neither requesters nor crowd workers were explicitly aware of each others' expertise or demographics. Workers did not know whether a requester was a designer, and requesters did not know crowd workers' backgrounds. The study moderators used the term "crowd workers" when providing the requesters study instructions, but did not provide other details about the particular crowd workers (e.g., demographics or skill levels) to avoid influencing the requesters' communication approaches. This reflects the temporary nature of current practice in crowdsourcing, where requesters do not know who crowd workers are [42].

4.3 Data Analysis

In order to capture the complete speech and canvas activity in Apparition, the audio and computer screens of T1, T2, and the follow-up interview were recorded. Both the audio extracted from the videos and the interviews were transcribed for data analysis. The transcribed speech from T1 and T2 was segmented into a series of speech utterances with a starting timestamp. The boundaries of each speech segment were drawn at pauses, at the end of sentences, or at a change in topic. Then, we developed and refined coding labels and interview questions over the course of pilot studies (eight runs) in the context of the initial research questions that we had (listed in the Introduction). Conversation coding has been used in previous studies that investigated shared visual contexts in collaborative design [15, 30, 34, 53]. We developed a set of coding labels as we observed the study, conducted interviews, and transcribed the recordings. In addition, through the process, we sought to identify emerging themes in relation to expertise, engagement, and temporal patterns. One of the authors coded all participant trials, and these coding results are what we present and analyze in sections 5 and 6. We also had another author code 10% of the data to report the inter-rater reliability of our coding. We calculated intraclass correlation coefficients (ICC) for the two coders' coding, and we report these scores and their interpretation (*poor, fair, good, excellent*) according to [9], following each of the label definitions below.

- ***Descriptive***: The requester describes how an element should look or describes the element's shape and geometrical properties (e.g., circle, rectangle, lines). (ICC : .746, *good*).
- ***Verbally Referential***: The segment contains words (including "deictic expressions") or phrases that refer to specific elements or a specific area of the canvas (e.g., "here", "the left side of the screen"). (ICC : .717, *good*).
- ***Diectic Gestures***: The requester uses the mouse cursor to point to a certain area or element on the canvas during the speech segment. (ICC : .834, $excellent$).
- ***Click Events***: The number of click events observed during the segment; during the task, click events were visualized with small circles on the participants' screens (Fig. 1-f). (ICC : .796, $excellent$).
- ***Corrective***: The requester asks workers to modify the existing elements they have created. (ICC : .547, $fair$).
- ***Repetitive***: The requester repeats a request that has been made earlier in the session. (ICC : .788, $excellent$).
- ***Feedback***: The requester gives positive/negative feedback to workers, including simple acknowledgements (e.g., "That looks good, thank you."). (ICC : .830, *excellent*).

Note that these labels are not necessarily mutually exclusive; a given segment can have more than one label. When coding, we also considered the chat history (Fig. 1-d) between crowd workers and the requester, which is interleaved with the audio recording of the requester and can influence the requester's spoken words.

4.4 Limitations

(In-lab Study) One limitation of our study is that participants were provided a pre-determined UI sketch to prototype, rather than allowed freedom to prototype a UI sketch of their own as is the end-goal of a system like Apparition. We opted for a task that balances flexibility and consistency, as having a controlled task avoids introduction of other factors that may affect requester communication style, for example requester UI sketches with differing levels of complexity. In addition, we found from the pilot studies that it was difficult for non-designers to come up with design ideas on the spot. A task was selected to simulate a hypothetical scenario in which participants have a conceptual idea in their mind. The study was carefully designed and finalized through eight pilot study sessions before the actual study. In coming up with design tasks, we consulted with two interaction designers and came up with 5 different scenarios, including an open-ended task that allows designers to have a lot of freedom for creation. The selected task included the simple, basic interactions common to mobile applications (e.g., drop-down menu, buttons, check and text boxes) but still be described in various ways. In general, we found that the task selected was a reasonable compromise for exploring the communication and collaboration process without having to implement the full pipeline of the system in the wild for a longitudinal study.
(Lack of Discussion on Performance) Although we present raw numbers for T2 task completion time and sketch-prototype quality, we do not attempt to quantitatively analyze how these results

correlate with requester communication style. The reason is that these results could be a compound outcome of many different factors, not only the ones that we are interested in for this study (i.e., a requester's communication with workers), but also workers' abilities (e.g., the skill level of workers in using the tool, communication and collaboration among workers) which are beyond the scope of this paper.

(Limited Number of Sessions and Workers) Lastly, it is noteworthy to mention that the number of sessions ($n = 10$) is not enough to quantitatively verify the findings. Rather, our study is primarily qualitative. While we cannot make comparative claims that require statistical significance (e.g., designers vs. non-designers, or T1 vs. T2), the qualitative analysis provides us with a strong understanding of the observed behaviors through interviews, conversation coding, and video review. Problematic incidents can be discovered with a small number of participants and can clearly inform how we can improve the current design of the system and similar future systems [50]. It is not uncommon to have a small scale (less than ten sessions) in such studies, especially for collaborative settings that involve more than one participant [3, 57, 58]. In addition, the number of crowd workers was typically four to five per session in the original Apparition study [38]. For our study, we recruited a smaller number of workers because having many workers may result in more worker-to-worker collaboration challenges (e.g., tasking, communication, self-coordination), which could complicate how the requester communicates.

5 RESULTS

We recorded completion times for both tasks per participant. The mean completion time for Task 1 (T1, without crowd workers) was 4.9 mins ($median = 4.5, \sigma = 2.2$). The mean completion time for Task 2 (T2, with crowd workers) was 19.7 mins ($median = 20, \sigma = 6.7$). Note that the completion times for the two tasks are not comparable, because T2 included the extra stage of completing the task with crowd workers that T1 did not. We evaluated the final crowd-created sketches in T2 against a rubric. The rubric measures recall of UI characteristics of the original sketch we provided to requesters. Across the ten T2 trials, the mean recall was 86% ($median = 94\%, \sigma = 15\%$). This result shows that a requester and crowd workers can effectively collaborate to create a medium-fidelity prototype in real time. The reported recall score can be considered effective, given that we did not provide requesters an explicit list of requirements alongside the sketch.

Based on the observations that we made from the transcriptions and video recordings, we present our findings in five different themes that emerged from the results and are relevant to the three motivational questions we presented in the introduction.

5.1 Real-time Collaboration with Workers in an Interactive Crowdsourcing System

In this section, we briefly discuss how the verbal communication patterns in Task 1 (T1, without crowd workers) and Task 2 (T2, with crowd workers) differ. In addition, we were interested to understand why and to what degree participants would or would not prefer this interactive crowdsourcing system over potential asynchronous collaboration.

5.1.1 ***Requesters' Involvement in Real-time Collaboration***. We observed that the requesters' speech and drawing activity in T2 is more sparse than in T1. The average number of words per minute (WPM) in T2 (avg=39 WPM, $\sigma = 15.6WPM$) is lower than it is in T1 (avg=75 WPM, $\sigma = 25.4$).[1]. This is not surprising, given that the requester in T2 was often silently monitoring the progress of the sketch and checking the elements that crowd workers created. Requesters then interrupted and gave both positive and negative feedback to crowd workers in response to the progress that they made, on average 13.2 times per session. For instance, "*I like the spacing in 'do*

[1]Both numbers are lower than the average (150 WPM) for a normal conversation [2].

laundry' between the check box" (P2), or "*no, no, no you can either get rid of these or put them over here*" (P9). In addition, requesters made corrective requests for crowd workers to modify existing elements, on average 10.6 times per session. For example, the comment by P9 above is a corrective comment. As such requests are typically made after the requester sees the actual elements that crowd workers have created, the corrections made with these comments would have resulted in incorrect elements if the crowd workers had worked asynchronously, as in T1. We looked in further detail at the corrective requests made in P9's T2, comparing them with the corresponding requests in their T1, and found out the corrective requests that P9 made were not simply due to the mistakes that workers made. Rather, we found that the majority of the corrective comments (76.7%) were 1) something that was visually implied but not explicitly mentioned in T1 (e.g., alignment, position) (38.3%), 2) something that was under-specified in T1 (16.8%), or 3) made on the fly only in T2 session (21.5%); the higher fidelity of the actual outcome made the requesters specify a new element property. Examples for such comments from participants in each category are as follows:

> (P1) "*Can you make this narrower? This is a mobile application but the screen looks like a desktop web page not mobile.*"
> (P2) "*If this text can be moved over to left aligned that would be great.*"
> (P9) "*Let's try to do a green check mark. (choosing the color of the check mark)*"

This indicates that the request made in T1 is not necessarily sufficient for crowd workers to produce exactly correct outcomes. Consistent with findings in collaborative design literature, the process of co-creation provides richer information to the participants than the resulting artifact [60] This clearly displays the benefits of real-time collaboration in crowd sourcing, at the price of more involvement in completing the task.

5.1.2 ***Participants' Reflection on Real-time Collaboration with Workers.*** In the follow-up interview, the participants were able to understand the benefits and challenges of real-time collaboration with workers in comparison to the asynchronous collaboration scenario. We asked the requesters which approach (T1: asynchronous vs. T2: real-time crowdsourcing) they would use in practice. 7 of 10 requesters preferred real-time collaboration over asynchronous collaboration. Requesters found "*benefits in being able to give feedback in real-time since the corrections could be made as they go rather than after*" (P6). P4's comment on this question highlights the benefits of real-time collaboration well:

> (P4) "*It's much less painstaking for me to give them live criticism than to make sure I talk out every detail in some sort of instructional video that I send off.*"

Three requesters (P2, P3, P5), all of whom are designers, indicated a preference to work asynchronously as they were in favor of being able to work in parallel. At the same time, they also understood why they, the requesters, might want to work in real time, reflecting their prior experiences in working asynchronously. P3 noted that working asynchronously is often followed by "*a lot of revision*" through multiple rounds of having to send a request, waiting for the task to be done, and then repeating this process, going back and forth until they are satisfied with the final result. The other two designers also expressed mixed feelings on their decisions.

> (P2) "*Because while I was watching, I did notice that I was feeling a little bit stressed when they got it wrong. But I also know that what happens sometimes is you just get it asynchronously, and that's also stressful, and that might even take more time, like explaining to them [workers] through email what's wrong, what needs to be corrected.*"
> (P5) "*It [watching it live] was like watching paint dry...But then for the actual people doing the task, if they don't understand something and you can't provide them feedback right away, I can see how that would also be frustrating for them. So, I don't know.*"

As illustrated by these quotes, the designers even exhibited frustration while watching workers working in real time. They might have underestimated the difficulties that non-expert crowd workers would face [24], especially given that the designers could have finished the prototype more quickly than the workers. A participant mentioned the potential of monitoring progress intermittently instead of in real time, and how such an approach might address their concerns on the continuous involvement with the benefits of having access to real-time progress on the work.

> (P2) "*I think I like being able to see the progress, but I would want to be able to multitask. I would like to have this somewhere playing as I'm doing my own work, and then I glance up at it and see it.*"

This suggests a hybrid method of both synchronous and asynchronous real-time interaction, and this mixed approach can lead to productivity gains, as two groups can work in parallel with less interruption [8]. In general, the majority of the participants preferred such interactive crowdsourcing systems, even though it could have been an unfamiliar experience of working with anonymous remote crowd workers.

5.2 Expertise, Language, and Jargon

Reviewing the specific language that requesters used to describe their sketches, it appears that the level of expertise determines the ways in which participants describe GUI elements. As the designers are familiar with a certain set of UI design terms, they speak differently from non-designers. For example, see the following examples of descriptions for the container window of the UI.

> (P7) "*So I am gonna have like a box here which is going to be an interface like a rectangular box, its height is bigger than the width.*"
> (P3) "*So to start, I am going to go ahead and draw the mobile device screen.*"

The example shows the non-designer (P7) characterized the element by its shape and geometric properties. In contrast, (P3) did not explicitly mention the shape of the element in their description.

In addition, non-designers tended to refer to elements in the canvas, using referential words or gestures with their mouse cursor. The participant (P7) in the example tried to provide visual context for their description, using diectic words such as "this" or "here", and associating the drawing with speech. On the other hand, the designer (P3) did not refer to the element on the canvas even though the participant was drawing the container on the canvas. This may be because designers have a clear connection between the spoken terms and the visuals on the canvas in their mental model. However, the crowd workers may not have the same clear connection, which could be a challenge and lead to a delay in comprehending the request for crowd workers [25]. To see if this communication style is consistent across participants, we counted the number of descriptions that are *descriptive* and *verbally referential*. Furthermore, we annotated the videos to see if they contained elements of *referring gestures* (e.g., a gesture of moving the mouse cursor in a small circle on top of an element) and *clicked* elements to generate click visualizations (Fig. 1-6). On average, the frequencies of descriptive expressions and reference to elements during the tasks were greater for non-designers than designers (Fig. 3). Similarly, non-designers tended to refer to elements with the mouse cursor (click or gesture) more than designers (Fig. 3). Prior work has shown that shared workspaces encourage users to rely on visual information to provide the necessary communicative and coordinating cues [14]. In our study, requesters without expertise seemed to have a stronger tendency of using visual cues, whereas requesters with expertise showed less reliance on them.

From a crowd worker's perspective, the same kind of unfamiliarity with UI design terminology might have been an obstacle in comprehending a designer's requests quickly, locating elements on the canvas, and catching implicit requirements. For example, when requesters asked workers to create the outer box of the application, the implicit requirement that the mobile phone screen

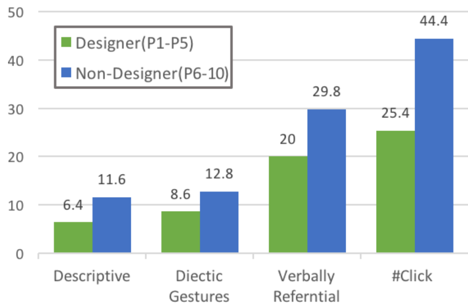

Fig. 3. On average, the non-designers described the request in more descriptive ways and referred to elements more frequently.

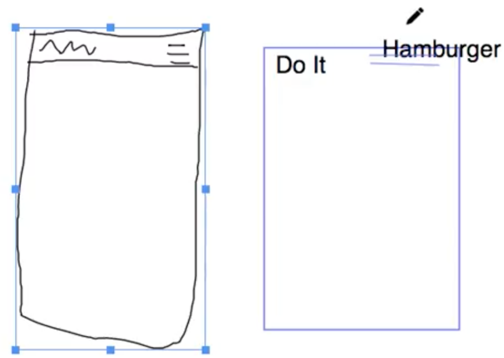

Fig. 4. *Hamburger menu gone wrong*: P1 used the jargon "hamburger menu" and a crowd worker spelled it out, as they were not familiar with the term.

be in "portrait" mode was missed by the workers in four out of ten sessions, and the requesters needed to correct them later. Another interesting example occurred when participants described the menu button, which is labeled as "drop down menu" in Fig. 2-(Given Sketch). The following are two examples from P8 and P1.

> (P8) "*I want to create like a drop down menu in the upper right-hand corner with like three little lines.*"
> (P1) "*I want you to add a hamburger menu on the top right corner.*"

In this quote, a designer (P1) and a non-designer (P8) described the same element in two distinct ways. Overall, three out of five designers (six out of eight if we include the designers from the pilot studies) used the jargon "hamburger menu", which refers to ☰. Zero non-designers used this jargon, and four out of five non-designers described the button based on its shape, with the phrase "three (parallel) lines". Using such jargon could lead to confusion if crowd workers are not familiar with a term and designers do not clarify it further. For example, a crowd worker in designer P1's session actually created a label that reads "hamburger" (Fig. 4.). This example illustrates the difficulties that non-expert crowd workers may have in comprehending an expert designer's descriptions. When there exists an expertise gap in dyadic communication, the process of assessing, supplying, and acquiring expertise is required to reach common ground [28]. However, the temporary and task-oriented nature of crowdsourcing seems not to foster this communication, and as a result, expert requesters may fail to realize that crowd workers sometimes have trouble comprehending their requests.

Arguably, this challenge can be mitigated if the system recruits expert crowd workers with design backgrounds from other crowdsourcing platforms, such as Upwork. However, if we recruit workers with expertise, the main benefits of crowdsourcing are likely to fade away; such expert workers may not be as cost-efficient, scalable, and available as a general online crowd. The current body of crowdsourcing research tends to keep the benefits and to close the gap between expert support and crowdsourcing by developing computational tools that mediate the crowd's efforts [6, 7, 38, 45, 49].

5.3 Speech, Text, Asymmetric Communication

As mentioned previously, using speech is essential to support multimodal interaction for a requester (speaking while drawing) [51]. The authors of Apparition chose to support multimodal inputs for the requester due to a set of advantages in expediting the creation process and capturing multiple information in a sketch concurrently. Indeed, the requesters in the study frequently performed mouse interactions (i.e., click, gesture, drawing elements, drawing to indicate) while they were speaking (48.4% of the total number of speech segments). In addition, requiring crowd workers to type responses instead of speaking them is necessary to allow requesters to make their verbal requests clearly, without interruption (e.g., a requester's voice can be easily lost in multiple crowd workers speaking simultaneously). Consequently, one of the unique characteristics of Apparition is the asymmetry of communication modalities, or cross-modal interaction between two groups: a requester speaks to workers and they respond by typing or making changes on the canvas. We discuss the challenges that arise due to asymmetric communication between the groups.

5.3.1 ***Lack of Immediate Responses.*** When we asked requesters *if they were bothered that they could speak to the workers while the workers could not speak to them*, 7 out of 10 of requesters were not in favor of asymmetric communication. Five of the participants expressed frustration at the lack of immediate responses from the workers when requests were made (P2, P3, P4, P5, P10). Three of them were not in favor of the inequality because they simply found it awkward and unnatural to have different modalities (P5, P6, P8). Conversing with the system without immediate responses may have made the communication feel one-sided, which can account for the reason why the participants thought it was awkward.

In general, delayed responses, both in communication and shared visual context of groupware, discourage participants from collaborating with each other [15, 19]. The challenge in Apparition seems to come from the asymmetric communication. Even when crowd workers responded in the chat box in text, the requesters often missed the textual confirmation from the workers. It seems that the requesters did not pay attention to the chatbox because the primary communication modality of a requester is speech. The chat box had a common design signifier to draw a user's attention to a new message; the title bar turns yellow when a new message is received, which goes away when the requester uses (or clicks on) it. However, as the requester can verbally respond, the highlight hardly ever went away, and the chat box generally remained constantly yellow, which does not help. Another study that has a similar setup also addressed the problem of participants missing textual comments from the workers [3].

On the other hand, lots of times there was no response at all; crowd workers started working without confirming their receipt of the request. However, starting to work on the request may not immediately provide any visual evidence that the request is being carried out through the synchronized canvas (e.g., navigating menu items) [20]. Reversely, it is likely that workers' textual messages may not receive a follow-up verbal response or simple acknowledgements from the requester or other audience members like in the case of live streaming services [52].

The absence of affirmative responses from the workers and the delay in having visual evidences may reduce overall workspace awareness in the system. Indeed, three requesters (P4, P7, P10) asked the moderator during the study if the workers were still there, having observed no activity both in the canvas and the chat box. P4's suggestion on having simple visual responses from the workers exemplifies their concern with the system:

> (P4) "*...you kind of wait to see 'Are they working on it? Did they not get it? Should I be typing this out?'. Yeah, it's too many what-if scenarios in my head. Like if there could even be a button that says 'Yep, got it', so they could say 'Cool, got it' in a single click, then I would feel better knowing that they've understood what I just said.*"

There occurred other relevant incidents that demonstrate the potential loss of having no responses. During T2 sessions, three requesters accidentally forgot to unmute the microphone when they were making requests. Due to the one-sided communication, it took a while for the participants to find out something was wrong and the mic was muted. For example, P9, who had muted the mic for 4.5 minutes, did not know for how long the crowd workers missed their verbal request(s). The participant expressed the challenge of having presence awareness when asked *if verbal communication was effective* during the interview.

> (P9) "*...I used filler words. Just as I didn't know that they were there, I kind of wanted them to know that I was there, even when I wasn't talking.*"

The participant verbally created an indication of presence for the crowd workers. This suggests the need to reinforce workspace awareness, as otherwise, unnecessary utterances may be used to remedy the perceived ignorance of the other group [10].

5.3.2 ***Mixed Ephemerality Creates Confusion.*** We observed that the mixed ephemerality of text and speech causes confusion in communication. In Apparition, chat messages are archived in the chat box so that both workers and requesters can browse the chat history. However, spoken words are ephemeral, leaving no record of having been uttered. Immediately, it was a challenge for crowd workers to recall all the spoken requests made earlier (Fig. 5-3). Some requesters (P2, P8) voluntarily chose to use the chat box, as typing the instructions leaves a record. This might have been because the crowd workers do not need to remember all the request details if it was specified in the chatbox. In addition, it was difficult to understand and recover the context from the chat history, as it only presents one side of the communication. For example, Fig. 5 reflects 9 minutes of conversation in P7's T2 session, in which a worker asked questions and responded to the requester. Apparently, parsing the chat history in retrospect provides almost no additional information, as the spoken parts are missing.

Furthermore, the archived textual history could add confusion, as it is not interleaved with the spoken instructions. For example, four consecutive "ok" or "yes" messages like in Fig. 5, can confuse a requester who may not know if the "yes" on the bottom was a response to their most recent request or an earlier one. When we annotated the video recordings of the study sessions, it was difficult for us to discern if the message was from a worker who just responded to a particular request or if it was from the past. The requesters should have had the similar challenges.

Questions that remain unanswered in the chat box also create unnecessary interaction. Three requesters (P2, P5, P7) verbally answered a question and chose to later leave a textual answer in the chat box. One of them (P5) provided the textual answer for a worker's question 70 seconds after they had verbally answered the question. The requester might have forgotten that they already gave an answer, or did not want to leave unanswered questions in the chat history. In any case, this created extra overhead for the requester physically and cognitively. From that point, they chose to use the chat box only and muted the microphone.

While we did not expect this asymmetric communication to be of interest in the study, we found a number of problematic incidents coming from this inequality. Live streaming services (e.g., Twitch, Facebook Live) also include cross-modal interaction (spoken and written), and yet are lacking analytic approaches [52]. We believe that addressing these challenges in general will help improve interactive crowdsourcing systems and other relevant live media.

5.4 Challenges in Co-working on a Shared Artifact

Having a shared visual context not only enables the remote co-creation of digital artifacts, but also enriches the communication and collaboration process [14, 15]. One unique challenge to an interactive crowdsourcing system is that the purpose of using the shared resource for requesters can be drastically different from the process for workers. Specifically, in this case, the rough sketch that a requester makes serves as a visual specification of the request (with corresponding narration) and the workers' drawing is the final outcome of the system.

During the process of converting a hand-drawn sketch into a prototype, we observed that workers often hesitated to replace pieces of the requester's drawing, even though during the

worker:*Can I delete the scribble?*
worker:*Ok*
worker:*And what was next? :D*
worker:*Ok*
worker:*Ok.*
worker:*Ok.*
worker:*Ok*
worker:*What was the next instructions?*

Fig. 5. 9 min-long history of chat during P7's session. 1) Browsing the chat history does not help recover the context. 2) The worker asks for permission to delete the requester's drawings (line 1). 3) The worker cannot recall the request (line 3 and 8).

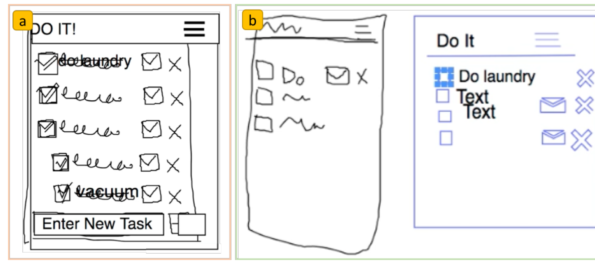

Fig. 6. a) *Crowded canvas*: The workers place their higher-fidelity square checkboxes and task label text directly on top of the corresponding elements hand-drawn by the requester, instead of deleting the hand-drawn elements first. b) *Split canvas*: The crowd workers choose to preserve the requester's drawing and create their prototype next to it.

training they were instructed to do so. In general, people naturally follow social protocols for mediating interactions, so it would have been unnatural for crowd workers to delete element drawings created by the requesters, even though they were instructed to do so in the interactive tutorial prior to the session [18]. Consequently, having two sets of the same element in varying fidelity (one hand-drawn by a requester, and the other UI element added by a crowd worker) can make the shared canvas cluttered quickly. We frequently observed cases where hand-drawn elements and their higher fidelity counterparts appeared overlaid, causing the canvas to be cluttered. Such cluttered elements not only make the sketch look unpolished, but also make it difficult to select requesters' elements that need to be removed eventually, because they are fully or partially obscured by the ones that crowd workers created later. P2's intermediate prototype in Fig. 6-a) shows an extreme case where the workers did not delete anything until the requester specifically asked them to. For example, after a worker replicated the outer border of the requester's mobile app sketch, the worker asked the requester for permission to delete the corresponding part of the requester's drawing (Fig. 5-2).

On the other hand, we noticed that removing pieces of the requester's initial sketch is problematic as well; it results in the loss of visual cues for recalling past requests and for understanding implicit requirements (e.g., alignment of multiple elements, spacing between elements) that may not be apparent when viewing individual elements of the sketch. In the T1 sessions of the study, the final outcome is a rough, but complete sketch; however, in the T2 sessions, usually workers do not have the opportunity to see a complete sketch like in Fig. 2-(T1).

Nearly half of the crowd team actively used a workaround to better handle this problem. In three of the sessions, crowd workers or requesters used a "sandbox" approach to demonstrate or brainstorm UI elements outside the canvas's whitespace and then place them in the right position when ready, with the removal of the corresponding sketched element immediately before the placement. This was effective in avoiding clutter and confusion with overlapping elements and minimizing the time that the requester's visual cues are missing. Another approach to avoid clutter, as well as to preserve the requester's hand-drawn sketch, was to use a split canvas. Crowd workers in two sessions used a split canvas, keeping the requester's drawing on one side of the screen and creating the higher fidelity prototype sketch on the other side of the screen (Fig. 6-b). In designing an interactive crowdsourcing system, it should be considered to keep the requesters' initial visual cues separate and protected from the crowd workers' outcome, as well as to make sure the protected visual cues do not physically interfere with workers' efforts on the task.

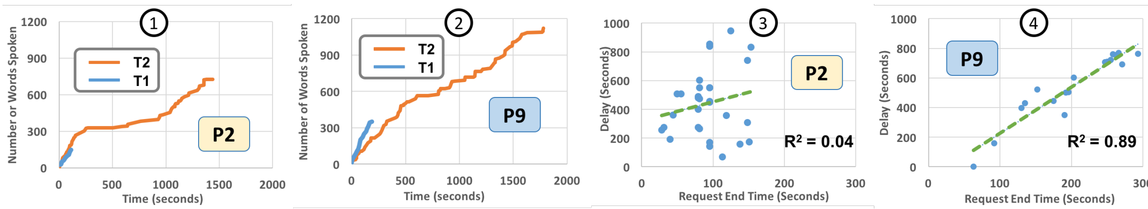

Fig. 7. **(1),(2)**: the number of words spoken so far **(y-axis)** over time **(x-axis)** for P2 (designer) and P9 (non-designer). The orange (longer) line indicates T2 and the blue (shorter) one indicates T1. For T2, P9 speaks in constant pace throughout the session, whereas P2 finished the initial requests quickly and stayed silent for a while. **(3),(4)**: response delay over time. **(x-axis)**: the timestamp at which an initial request was made per element, **(y-axis)**: the time (delay) it took for crowd workers to respond to the request. Delay for P9 showed a linear trend, which indicates the requests got quickly backlogged.

5.5 Varying Pace of Making Requests

In this kind of collaboration, requesters may communicate their UI requirements to workers in temporally different fashions. Examining the variability of speech pace can help us understand temporal patterns of how they make requests over time during the session and see if there exists any challenge involved. To investigate the temporal pattern of a speaker, we calculated a participant's pace (speech rate) and saw how it changed over time for all participants. Here, we present two participants' speech rates, showing two different representative patterns in Fig 7-1 and Fig 7-2; the slope of the graph indicates speech rate of the person in the particular task (words per minute). When P2 conducted T2 (orange line in Fig 7-1), they started speaking at an approximately constant pace for the first 154 seconds, closely matching the pace when they did T1 (the blue line of the same figure). P2 then stayed mostly silent for 10 minutes thereafter (154—764 seconds in Fig 7-1.), with only a few spoken clarifications, allowing the workers to catch up on completing requests. In the meantime, P9 spoke at an approximately constant words-per-minute rate throughout their T2 session, as can be seen by the approximately linear orange line in Fig 7-2. One potential reason that can account for the difference between P2 and P9 could come from their varying levels of expertise. Reviewing screen recordings in detail, we found that P2, a UI designer with 4 years of professional experience, was effective and efficient in describing and creating a rough sketch of the given UI (e.g., better/quicker drawing, extensive use of copy/paste which led to consistency between elements of same type). The sketch gave the workers a complete visual specification that demonstrates the overall structure of the application effectively. In the meantime, P9, a non-designer, manually drew each element of the interface, taking longer time to complete the sketch than P2.

The level of effectiveness of a requester's description and drawing can also influence the workers' request completion patterns. For example, Fig. 7-3 and Fig. 7-4 illustrate the delay from the time that a participant (P2 or P9) made the initial request for an element to the time that a worker began to work on that particular element in T2. For example, if the coordinate of a data point is (100,20), it means that the crowd workers began working on the corresponding element 20 seconds after it was requested at time 100 seconds. We present data for the first five minutes of each session (P2 and P9), as this corresponds to the time that the bulk of initial element creation requests were made, and most of the remaining communications thereafter were clarifications about the initial requests or corrections to the prototype. The graph for P9 (Fig. 7-4) shows a fairly linear trend between the time that a request was made and the delay for that element, with an $R^2 = 0.89$, indicating many requests got backlogged over time. The graph for P2 (Fig. 7-3), however, shows no such trend; delay for response to a request seems to have no correlation with the time a request was made. Rather, the same kinds of elements (i.e., check-boxes, email icons) have been created with similar delays.

Potential challenges that crowd workers would have had in P9's session include needing to listen to the request as well as working on the prototype and not being able to recall all the requests that were made later in the session as the delay gets longer. This sometimes forced a requester to make the same request again. On average, the participants made *repetitive* requests 5.0 times during their T2 sessions. On the other hand, crowd workers also asked requesters to repeat the instructions from time to time (1.4 times per session on average; see the example message in Fig. 5-3). Overall, we can see that varying temporal patterns can pose challenges for crowd workers to respond to the request timely and can impact crowd workers' response time within a session. It would be worthwhile to consider how such a system can address these challenges for workers, and to design the system to guide requesters to effectively make requests.

6 DESIGN RECOMMENDATIONS

Based on the insights we learn from the study, we suggest design recommendations for crowd-powered design tools that involve requesters and crowd workers. We believe these will be applicable to interactive crowdsourcing systems that have a similar structure of user-crowd collaboration, especially in creative domains.

6.1 Turn Live Speech into a Structured Task List

Due to the asymmetry in communication, requesters will not have a fluent communication close to in-person collaboration and remote conference calls. A backlog of requests can build up quickly if a requester chooses to interleave their requests with giving feedback on and correcting existing artifacts. To that end, crowd workers need ways to structure verbal requests and to revisit past requests. It will be desirable for crowd workers to be able to replay the audio for a specific task. In addition, workers should be able to quickly find and navigate what to replay close to a hierarchical to-do list where each item is associated with an audio snippet, as suggested in [44]. This may help requesters reduce the time they spend monitoring the system in real time; workers can replay the requester's verbal instructions and, therefore, hopefully complete more tasks the first time they are asked, with fewer clarifying questions and less repetitive requests. Given the real-time audio streaming, this poses an interesting design challenge of recording live audio as well as being able to revisit the audio at the same time because the two actions are usually mutually exclusive; audio being replayed would mask the live audio.

6.2 Address Asymmetry in Communication and Expertise

We found that the asymmetric communication pattern of a requester's speech versus the workers' typed text poses a number of challenges. We believe there are certain cases where this asymmetry is inevitable due to the scale (e.g., live streaming events in social media—Twitch, Facebook Live), but the slowdown of communication in a collaborative setting can be costly in crowd-powered systems. An easy but overly-simplistic solution to this problem could be to match modalities automatically, when possible. For example, spoken communication could be transcribed into the chat box. Further, the developers of such systems should be aware that the potential expertise gap can be a challenge in communication when designing interactive tutorials and recruiting workers (relevant background, prior experience with the task). Providing general guidelines and rubric can reconcile the expertise gap and raise the perceived value of a crowd worker's outcome as found in [62]. More advanced NLP techniques that can adapt the reference to more general terms can be anticipated in interactive crowdsourcing systems [29]. We believe the asymmetry in communication and expertise is an intriguing research topic that requires separate attention in future work.

6.3 Protect Requester Artifacts in Shared Space

As previously mentioned, state sharing in crowd-powered systems should occur in different ways, especially given that requesters and workers have different purposes with regard to the shared artifact. In our case, a requester's sketch being deleted bit by bit over time, as well as the canvas being cluttered, were problematic. Social protocols in crowdsourcing systems can be vague, given the temporary and remote nature of the process. This necessitates coordination policies embedded in the system [48]. Particularly, in this design context, we identified two needs for future systems: 1) the requester's visual cues should remain intact and protected (access control), and 2) the requester's visual cues should not interfere with the workers' process.

A simple solution for this is to denote separate prototyping areas side by side on the canvas, one for the requester only and one for the workers only, which is similar to the strategy employed by some crowd workers who split the canvas during their session [12]. Giving different read-write access to requesters and workers will help protect the requesters' visual cues. Alternatively, a requester's sketch can be visually presented as a transparent overlay on the workers' canvas, with functionality to easily remove or hide it.

6.4 Augment Work-space Awareness of Presence and Implicit Interactions

It seems that the workspace awareness is much more limited, not only because the task involves remote collaboration, but also because of the transient nature of crowdsourcing and the asymmetrical communication [20]. Even with the shared canvas and live audio streaming, requesters had trouble in using the system. For example, during the experiment, requesters wanted to ensure their voices were audible to crowd workers. This can be challenging, again due to the lack of immediate (audible) feedback from the crowd workers when they do not respond through the chat box quickly. A simple visualization of the requester's audio as heard on the workers' machines can help reassure the requester that audio is functioning properly on all team members' machines. We also found that non-designers referred to elements by naturally gesturing with their mouse cursors, though not clicking. The deictic gestures made with the mouse cursor did not help crowd workers locate an element, because crowd workers cannot see the mouse cursor's movement unless the mouse is clicked; this information could be useful to crowd workers if visualized in sync with the speech. Gesture traces can be used to supplement the shared visual space [21]. As the canvas is synchronized per-element (as opposed to per-pixel), visualizing the activity level of a worker can enhance awareness of whether they are working on something or are idle. Lastly, crowdsourcing systems are often designed to intentionally mask existing information for privacy, in which case developers need to consider how to preserve awareness in such systems, especially if they should support real-time collaboration [31, 37].

6.5 Train Workers to Collaborate, Not Just Use

We believe that the findings from this work should be considered when training crowd workers. Until now, the training for systems like Apparition has focused on tool usage. However, there are other aspects that facilitate fluid collaboration for crowd workers to understand, for example, acknowledging requests by responding immediately and addressing the requests in a timely manner. In addition, there are other important factors that this study did not explore but are essential to accomplish the goal of real-time collaboration, such as task coordination, awareness of tasking, and communication amongst workers. Designing interactive tutorials that teach these concepts will help crowd workers be effective in real-time collaboration.

7 CONCLUSION

In this paper, we explored the collaborative aspects of a real-time interactive crowdsourcing system and aimed to better understand the interaction between requesters and the crowd workers who support such systems. We conducted a user study that allowed us to understand how real-time collaboration occurs and how requesters understand such a collaboration type. We also explored various aspects of requester-workers collaboration: the style of language they speak and corresponding challenges, the ways in which they work together on a shared canvas, the challenges arising from asymmetric communication channels, and the variability of their communication patterns over time. Our study's findings directly inform the design of future crowd-powered prototyping systems, and, more generally, will help reinforce our understanding of the interaction between requesters and online crowd workers, especially in the domain of live collaborative creation [43]. We hope that our design recommendations we make will help provide a reference for researchers and practitioners who are interested in developing more powerful interactive crowdsourcing systems, as well as mixed-role interactive systems more broadly.

ACKNOWLEDGMENTS

We thank our reviewers for their constructive feedback on this paper. We also thank our study participants for their time and effort. This work was supported by Rackham Graduate School, University of Michigan, MCubed project, and IBM.

REFERENCES

[1] 2017. Amazon's Mechanical Turk. http://www.mturk.com.
[2] 2017. Voice Qualities. http://www.ncvs.org/ncvs/tutorials/voiceprod/tutorial/quality.html.
[3] Salvatore Andolina, Hendrik Schneider, Joel Chan, Khalil Klouche, Giulio Jacucci, and Steven Dow. 2017. Crowdboard: Augmenting In-Person Idea Generation with Real-Time Crowds. In *Proceedings of the 2017 ACM SIGCHI Conference on Creativity and Cognition (C&C '17)*. ACM, New York, NY, USA, 106–118. https://doi.org/10.1145/3059454.3059477
[4] Adam Bergkvist, Daniel C Burnett, Cullen Jennings, and Anant Narayanan. 2012. Webrtc 1.0: Real-time communication between browsers. *Working draft, W3C* 91 (2012).
[5] Michael S. Bernstein, Joel Brandt, Robert C. Miller, and David R. Karger. 2011. Crowds in Two Seconds: Enabling Realtime Crowd-powered Interfaces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 33–42. https://doi.org/10.1145/2047196.2047201
[6] Michael S. Bernstein, Greg Little, Robert C. Miller, Björn Hartmann, Mark S. Ackerman, David R. Karger, David Crowell, and Katrina Panovich. 2015. Soylent: A Word Processor with a Crowd Inside. *Commun. ACM* 58, 8 (July 2015), 85–94. https://doi.org/10.1145/2791285
[7] Jeffrey P. Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C. Miller, Robin Miller, Aubrey Tatarowicz, Brandyn White, Samual White, and Tom Yeh. 2010. VizWiz: nearly real-time answers to visual questions. In *User Interface Software and Technology (UIST)*. 333–342. https://doi.org/10.1145/1866029.1866080
[8] Yan Chen, Sang Won Lee, Yin Xie, Yiwei Yang, Walter S. Lasecki, and Steve Oney. 2017. Codeon: On-Demand Software Development Assistance. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 6220–6231. https://doi.org/10.1145/3025453.3025972
[9] Domenic V Cicchetti. 1994. Guidelines, criteria, and rules of thumb for evaluating normed and standardized assessment instruments in psychology. In *Psychological Assessment, 6(4)*. 284–290.
[10] Herbert H Clark, Susan E Brennan, et al. 1991. Grounding in communication. *Perspectives on socially shared cognition* 13, 1991 (1991), 127–149.
[11] Steven Dow, Anand Kulkarni, Scott Klemmer, and Björn Hartmann. 2012. Shepherding the Crowd Yields Better Work. In *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work (CSCW '12)*. ACM, New York, NY, USA, 1013–1022. https://doi.org/10.1145/2145204.2145355
[12] Clarence A. Ellis, Simon J. Gibbs, and Gail Rein. 1991. Groupware: Some Issues and Experiences. *Commun. ACM* 34, 1 (Jan. 1991), 39–58. https://doi.org/10.1145/99977.99987
[13] Susan R. Fussell, Robert E. Kraut, and Jane Siegel. 2000. Coordination of Communication: Effects of Shared Visual Context on Collaborative Work. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*

(CSCW '00). ACM, New York, NY, USA, 21–30. https://doi.org/10.1145/358916.358947

[14] Darren Gergle, Robert E. Kraut, and Susan R. Fussell. 2004. Action As Language in a Shared Visual Space. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work (CSCW '04)*. ACM, New York, NY, USA, 487–496. https://doi.org/10.1145/1031607.1031687

[15] Darren Gergle, Robert E. Kraut, and Susan R. Fussell. 2004. Language Efficiency and Visual Technology: Minimizing Collaborative Effort with Visual Information. *Journal of Language and Social Psychology* 23, 4 (2004), 491–517. https://doi.org/10.1177/0261927X04269589 arXiv:https://doi.org/10.1177/0261927X04269589

[16] Mitchell Gordon, Jeffrey P. Bigham, and Walter S. Lasecki. 2015. LegionTools: A Toolkit + UI for Recruiting and Routing Crowds to Synchronous Real-Time Tasks. In *Adjunct Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15 Adjunct)*. ACM, New York, NY, USA, 81–82. https://doi.org/10.1145/2815585.2815729

[17] Sai R. Gouravajhala, Jinyeong Yim, Karthik Desingh, Yanda Huang, Odest Chadwicke Jenkins, and Walter S. Lasecki. 2018. EURECA: Enhanced Understanding of Real Environments via Crowd Assistance. In *Proceedings of the Sixth AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2018, Zürich, Switzerland, July 5-8, 2018*. 31–40. https://aaai.org/ocs/index.php/HCOMP/HCOMP18/paper/view/17933

[18] Saul Greenberg and David Marwood. 1994. Real Time Groupware As a Distributed System: Concurrency Control and Its Effect on the Interface. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work (CSCW '94)*. ACM, New York, NY, USA, 207–217. https://doi.org/10.1145/192844.193011

[19] Carl Gutwin. 2001. The effects of network delays on group work in real-time groupware. In *ECSCW 2001*. Springer, 299–318.

[20] Carl Gutwin and Saul Greenberg. 2002. A descriptive framework of workspace awareness for real-time groupware. *Computer Supported Cooperative Work (CSCW)* 11, 3-4 (2002), 411–446. https://doi.org/10.1023/A:1021271517844

[21] Carl Gutwin and Reagan Penner. 2002. Improving Interpretation of Remote Gestures with Telepointer Traces. In *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work (CSCW '02)*. ACM, New York, NY, USA, 49–57. https://doi.org/10.1145/587078.587086

[22] Oliver L. Haimson and John C. Tang. 2017. What Makes Live Events Engaging on Facebook Live, Periscope, and Snapchat. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 48–60. https://doi.org/10.1145/3025453.3025642

[23] William A. Hamilton, Oliver Garretson, and Andruid Kerne. 2014. Streaming on Twitch: Fostering Participatory Communities of Play Within Live Mixed Media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 1315–1324. https://doi.org/10.1145/2556288.2557048

[24] Pamela J Hinds. 1999. The curse of expertise: The effects of expertise and debiasing methods on prediction of novice performance. *Journal of Experimental Psychology: Applied* 5, 2 (1999), 205.

[25] Pamela J Hinds, Michael Patterson, and Jeffrey Pfeffer. 2001. Bothered by abstraction: The effect of expertise on knowledge transfer and subsequent novice performance. *Journal of applied psychology* 86, 6 (2001), 1232.

[26] Simo Hosio, Jorge Goncalves, Niels van Berkel, Simon Klakegg, ShinâĂŹIchi Konomi, and Vassilis Kostakos. 2017. Facilitating Collocated Crowdsourcing on Situated Displays. *Human-Computer Interaction* 0, 0 (2017), 1–37. https://doi.org/10.1080/07370024.2017.1344126 arXiv:https://doi.org/10.1080/07370024.2017.1344126

[27] Ting-Hao Kenneth Huang, Walter S. Lasecki, Amos Azaria, and Jeffrey P Bigham. 2016. "Is There Anything Else I Can Help You With?" Challenges in Deploying an On-Demand Crowd-Powered Conversational Agent. In *Fourth AAAI Conference on Human Computation and Crowdsourcing*.

[28] Ellen A. Isaacs and Herbert H. Clark. 1987. References in conversation between experts and novices. *Journal of experimental psychology: general* 116, 1 (1987), 26.

[29] Srinivasan Janarthanam and Oliver Lemon. 2010. Learning to adapt to unknown users: referring expression generation in spoken dialogue systems. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 69–78.

[30] Laurent Karsenty. 1999. Cooperative work and shared visual context: An empirical study of comprehension problems in side-by-side and remote help dialogues. *Human-Computer Interaction* 14, 3 (1999), 283–315. http://dx.doi.org/10.1207/S15327051HCI1403_2

[31] Harmanpreet Kaur, Mitchell Gordon, Yiwei Yang, Jeffrey P. Bigham, Jaime Teevan, Ece Kamar, and Walter S. Lasecki. 2017. Crowdmask: Using crowds to preserve privacy in crowd-powered systems via progressive filtering. In *Proceedings of the AAAI Conference on Human Computation (HCOMP 2017)., HCOMP*, Vol. 17.

[32] Aniket Kittur, Jeffrey V. Nickerson, Michael Bernstein, Elizabeth Gerber, Aaron Shaw, John Zimmerman, Matt Lease, and John Horton. 2013. The Future of Crowd Work. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work (CSCW '13)*. ACM, New York, NY, USA, 1301–1318. https://doi.org/10.1145/2441776.2441923

[33] Aniket Kittur, Boris Smus, Susheel Khamkar, and Robert E. Kraut. 2011. CrowdForge: Crowdsourcing Complex Work. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 43–52. https://doi.org/10.1145/2047196.2047202

[34] Robert E. Kraut, Susan R. Fussell, and Jane Siegel. 2003. Visual Information as a Conversational Resource in Collaborative Physical Tasks. *Human-Computer Interaction* 18, 1-2 (2003), 13–49. https://doi.org/10.1207/S15327051HCI1812_2

[35] Walter S. Lasecki and Jeffrey P Bigham. 2013. Interactive crowds: Real-time crowdsourcing and crowd agents. In *Handbook of human computation*. Springer, 509–521.

[36] Walter S. Lasecki, Mitchell Gordon, Danai Koutra, Malte F. Jung, Steven P. Dow, and Jeffrey P. Bigham. 2014. Glance: Rapidly Coding Behavioral Video with the Crowd. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 551–562. https://doi.org/10.1145/2642918.2647367

[37] Walter S. Lasecki, Mitchell Gordon, Winnie Leung, Ellen Lim, Jeffrey P. Bigham, and Steven P. Dow. 2015. Exploring Privacy and Accuracy Trade-Offs in Crowdsourced Behavioral Video Coding. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1945–1954. https://doi.org/10.1145/2702123.2702605

[38] Walter S. Lasecki, Juho Kim, Nick Rafter, Onkur Sen, Jeffrey P. Bigham, and Michael S. Bernstein. 2015. Apparition: Crowdsourced User Interfaces That Come to Life As You Sketch Them. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1925–1934. https://doi.org/10.1145/2702123.2702565

[39] Walter S. Lasecki, Kyle I. Murray, Samuel White, Robert C. Miller, and Jeffrey P. Bigham. 2011. Real-time Crowd Control of Existing Interfaces. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 23–32. https://doi.org/10.1145/2047196.2047200

[40] Walter S. Lasecki, Phyo Thiha, Yu Zhong, Erin Brady, and Jeffrey P. Bigham. 2013. Answering Visual Questions with Conversational Crowd Assistants. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '13)*. ACM, New York, NY, USA, Article 18, 8 pages. https://doi.org/10.1145/2513383.2517033

[41] Walter S. Lasecki, Rachel Wesley, Jeffrey Nichols, Anand Kulkarni, James F. Allen, and Jeffrey P. Bigham. 2013. Chorus: A Crowd-powered Conversational Assistant. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 151–162. https://doi.org/10.1145/2501988.2502057

[42] Matthew Lease, Jessica Hullman, Jeffrey P. Bigham, Michael S. Bernstein, Juho Kim, Walter S. Lasecki, Saeideh Bakhshi, Tanushree Mitra, and Robert C Miller. 2013. Mechanical turk is not anonymous. (2013). http://dx.doi.org/10.2139/ssrn.2228728

[43] Sang Won Lee. 2018. *Improving User Involvement through live collaborative creation.* Ph.D. Dissertation. University of Michigan, Ann Arbor, MI, USA.

[44] Sang Won Lee, Yan Chen, and Walter S Lasecki. 2017. The Need for Real-Time Crowd Generation of Task Lists from Speech. In *Work-in-Progress, the AAAI Conference on Human Computation (HCOMP 2017).*

[45] Sang Won Lee, Y. Zhang, I. Wong, Yang Y., S. O'Keefe, and W.S. Lasecki. 2017. SketchExpress: Remixing Animations For More Effective Crowd-Powered Prototyping Of Interactive Interfaces. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*. ACM. https://doi.org/10.1145/3126594.3126595

[46] Greg Little, Lydia B. Chilton, Max Goldman, and Robert C. Miller. 2009. TurKit: Tools for Iterative Tasks on Mechanical Turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation (HCOMP '09)*. ACM, New York, NY, USA, 29–30. https://doi.org/10.1145/1600150.1600159

[47] Alan Lundgard, Yiwei Yang, Maya L. Foster, and Walter S. Lasecki. 2018. Bolt: Instantaneous Crowdsourcing via Just-in-Time Training. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 467, 7 pages. https://doi.org/10.1145/3173574.3174041

[48] Meredith Ringel Morris, Kathy Ryall, Chia Shen, Clifton Forlines, and Frederic Vernier. 2004. Beyond "Social Protocols": Multi-user Coordination Policies for Co-located Groupware. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work (CSCW '04)*. ACM, New York, NY, USA, 262–265. https://doi.org/10.1145/1031607.1031648

[49] Michael Nebeling, Alexandra To, Anhong Guo, Adrian A. de Freitas, Jaime Teevan, Steven P. Dow, and Jeffrey P. Bigham. 2016. WearWrite: Crowd-Assisted Writing from Smartwatches. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 3834–3846. https://doi.org/10.1145/2858036.2858169

[50] Jakob Nielsen and Thomas K. Landauer. 1993. A Mathematical Model of the Finding of Usability Problems. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems (CHI '93)*. ACM, New York, NY, USA, 206–213. https://doi.org/10.1145/169059.169166

[51] Sharon Oviatt, Phil Cohen, Lizhong Wu, Lisbeth Duncan, Bernhard Suhm, Josh Bers, Thomas Holzman, Terry Winograd, James Landay, Jim Larson, and David Ferro. 2000. Designing the User Interface for Multimodal Speech and Pen-Based Gesture Applications: State-of-the-Art Systems and Future Research Directions. *Human-Computer Interaction* 15, 4 (2000), 263–322. https://doi.org/10.1207/S15327051HCI1504_1

[52] Daniel Recktenwald. 2017. Toward a transcription and analysis of live streaming on Twitch. *Journal of Pragmatics* 115 (2017), 68 – 81. https://doi.org/10.1016/j.pragma.2017.01.013

[53] Fraser J. M. Reid and Susan E. Reed. 2007. Conversational grounding and visual access in collaborative design. *CoDesign* 3, 2 (2007), 111–122. https://doi.org/10.1080/15710880601143195

[54] Daniela Retelny, Sébastien Robaszkiewicz, Alexandra To, Walter S. Lasecki, Jay Patel, Negar Rahmati, Tulsee Doshi, Melissa Valentine, and Michael S. Bernstein. 2014. Expert Crowdsourcing with Flash Teams. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 75–85. https://doi.org/10.1145/2642918.2647409

[55] Niloufar Salehi, Andrew McCabe, Melissa Valentine, and Michael Bernstein. 2017. Huddler: Convening Stable and Familiar Crowd Teams Despite Unpredictable Availability. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW '17)*. ACM, New York, NY, USA, 1700–1713. https://doi.org/10.1145/2998181.2998300

[56] Niloufar Salehi, Jaime Teevan, Shamsi Iqbal, and Ece Kamar. 2017. Communicating Context to the Crowd for Complex Writing Tasks. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW '17)*. ACM, New York, NY, USA, 1890–1901. https://doi.org/10.1145/2998181.2998332

[57] Stacey D. Scott, M. Sheelagh T. Carpendale, and Kori M. Inkpen. 2004. Territoriality in Collaborative Tabletop Workspaces. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work (CSCW '04)*. ACM, New York, NY, USA, 294–303. https://doi.org/10.1145/1031607.1031655

[58] Anthony Tang, Michel Pahud, Kori Inkpen, Hrvoje Benko, John C. Tang, and Bill Buxton. 2010. Three's Company: Understanding Communication Channels in Three-way Distributed Collaboration. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work (CSCW '10)*. ACM, New York, NY, USA, 271–280. https://doi.org/10.1145/1718918.1718969

[59] John C Tang. 1989. Listing, drawing and gesturing in design: A study of the use of shared workspaces by design teams. *Ph.D. Thesis* (1989).

[60] John C Tang. 1991. Findings from observational studies of collaborative work. *International Journal of Man-machine studies* 34, 2 (1991), 143–160.

[61] Melissa A. Valentine, Daniela Retelny, Alexandra To, Negar Rahmati, Tulsee Doshi, and Michael S. Bernstein. 2017. Flash Organizations: Crowdsourcing Complex Work by Structuring Crowds As Organizations. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 3523–3537. https://doi.org/10.1145/3025453.3025811

[62] Alvin Yuan, Kurt Luther, Markus Krause, Sophie Isabel Vennix, Steven P Dow, and Bjorn Hartmann. 2016. Almost an Expert: The Effects of Rubrics and Expertise on Perceived Value of Crowdsourced Design Critiques. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing (CSCW '16)*. ACM, New York, NY, USA, 1005–1017. https://doi.org/10.1145/2818048.2819953

Received April 2018; revised July 2018; accepted September 2018