Continual Support: An Approach for Engaging Ad-Hoc Crowds in Community Support

LEESHA V. MALIAKAL, Northwestern University
HAOQI ZHANG, Northwestern University

UPDATED ON 2020-01-16 06:08:07-06:00

Individuals in groups and communities are often willing, available, and able to support one another's needs. However, these individuals often struggle to balance opportunities to support the needs of others with focusing on their own goals when needs and goals change continuously. We contribute continual support – a conceptual approach and technical framework that monitors emerging needs and the status of individuals' goals, matches individuals to opportune moments to support others' needs or focus on their own goals, and primes individuals with the necessary context to provide support on-demand between goals. We demonstrate the value of the continual support framework through CrowdCheer, a mobile application to connect an ad-hoc crowd of spectators at a race to opportunities to cheer for other runners while still cheering for their own runner. A pilot study at a 10K race demonstrates how implementing a continual support approach enabled an ad-hoc crowd to balance supporting the needs of others with focusing on their own goals with ease. Results show that spectators were able to balance needs and goals because they maintained awareness of others' needs and their own goals; effectively evaluated when to support others' needs vs. attend to their own goals; and were prepared to support emerging needs on-demand between goals, which they valued. These findings provide early evidence for how CSCW technologies capable of continuously monitoring and connecting emerging needs to available resources can improve and scale how individuals in groups and communities can support one another while still meeting their own goals.

CCS Concepts: • **Human-centered computing → Interactive systems and tools**; **Collaborative and social computing systems and tools**;

Additional Key Words and Phrases: continual support, continual support systems, continual support framework, ad-hoc crowds

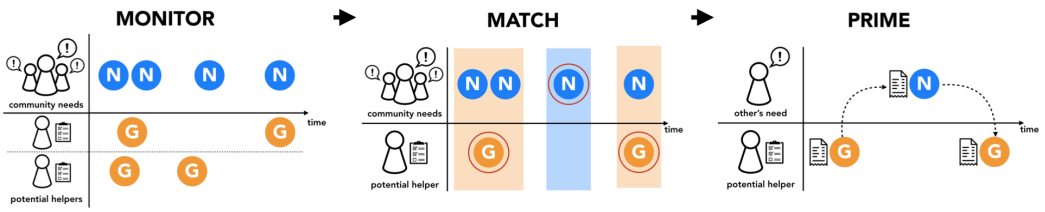
ACM Reference Format:
Leesha V. Maliakal and Haoqi Zhang. 2020. Continual Support: An Approach for Engaging Ad-Hoc Crowds in Community Support. *J. ACM* 37, 4, Article 111 (January 2020), 25 pages. https://doi.org/10.1234/0000000.0000000

1 INTRODUCTION

People in group and community settings are often willing, available, and able to support the emerging needs of others between their own goals. We see this in work and learning communities, like when a graduate student is willing and has the domain expertise to offer feedback on a peer's study design between their own research deadlines. We also see this in some crowd settings, like when a spectator at a race is willing and has time to cheer for other runners while they wait for their own runner to race past. These individuals are well-equipped to help because they may value

Authors' addresses: Leesha V. Maliakal, Northwestern University, lmaliakal@u.northwestern.edu; Haoqi Zhang, Northwestern University, hq@northwestern.edu.

0004-5411/2020/1-ART111 $15.00
https://doi.org/10.1234/0000000.0000000



(a) (b) (c)

Fig. 1. We contribute **continual support**, a conceptual approach and technical framework that directly enables potential helpers to balance supporting community needs (N) with focusing on their own goals (G). The continual support framework (a) *monitors* the changing status of people's goals and emerging needs; (b) *matches* individuals to opportune moments to support others' needs or focus on their goals; and (c) *primes* individuals with the context needed to provide on-demand support between goals.

supporting the types of needs that arise in the community, have moments of availability to offer support between their own goals, and have the relevant skills or prior experience to effectively help. More than ever, the development of better communication tools [32], expertise recommenders [26], matching algorithms [28], and real-time [5] and volunteer [21] crowd support systems have made it easier to connect individuals who need support to such potential helpers.

However, the moments when community needs arise and a potential helper's goals demand attention *change continuously*, and can occur at the *same time*. In these situations, potential helpers can struggle to balance a desire to support community needs with focusing on their own goals. For example, while a graduate student may be generally willing and have the domain expertise to help a peer with their study design, in the moment they may struggle to recognize they do not have the capacity to help that week given their own upcoming conference deadline. Similarly, while a spectator may be generally willing and available to cheer for other runners at a race, they may hesitate to do so if they are focused on finding their own runner in the crowd, or may be unprepared to cheer on the fly for someone they do not know. Despite being well-equipped to offer support, such potential helpers can struggle to (a) track moments when needs arise and goals demand attention; (b) decide when they have the capacity to support others' needs or need to focus on their own goals; and (c) prepare to support those needs on-demand between their goals. Consequently, potential helpers may refrain from supporting needs that they have the capacity to support, or when they do not have the capacity, may provide help at the cost of their own goals.

While CSCW and crowd computing research have explored ways to connect needs to helpers in communities and enable on-demand support, there exists a gap in approaches that directly enable potential helpers to balance supporting emerging needs of others with focusing on their own goals. Existing expertise recommendation systems (e.g. [26]) often lack models of the changing status of people's needs and goals, which are needed to connect a person's need to an available helper in a given moment. Prior work in real-time crowd support (e.g. [5]) often does not consider settings where individuals are volunteering time to support others' needs between their own goals, which can change their capacity and willingness to contribute support in the moment. Addressing these gaps requires integrated approaches that can monitor emerging needs and the changing status of goals in communities, and coordinate support in ways that are cognizant of and directly support the goals of those volunteering support.

To overcome these challenges, we contribute ***continual support*** – a conceptual approach and technical framework that directly enables potential helpers to balance supporting community needs with focusing on their own goals (Figure 1). To enable potential helpers to maintain awareness of the moments where others' needs arise and their own goals demand attention, continual support *monitors* community needs and the goals of potential helpers explicitly (Figure 1a). Continual support then *matches* potential helpers to opportune moments to support needs or focus on their own goals (Figure 1b), thus shifting the focus of individuals to their own goals as they demand their attention (Figure 1b, orange panels); and towards emerging needs when they have the capacity to help (Figure 1b, blue panel). To prepare individuals for the moments where needs arise and goals demand attention, continual support then *primes* helpers with the context they need to provide on-demand support and fluidly transition back to their own goals (Figure 1c). By directly enabling individuals to maintain awareness of, evaluate, and prepare for the moments when they have the capacity to support needs *and* the moments to focus on their own goals, continual support has the potential to sustainably expand needs that can be met with existing community resources.

To demonstrate the value of continual support, we study its use for engaging ***ad-hoc crowds*** – people who may not have explicit social ties to those who need help, but while engaged in a primary activity are generally willing, available, and able to provide support to others at opportune moments. We focus on the practical example of an ad-hoc crowd of spectators at a race, and implement the continual support framework in *CrowdCheer*, a mobile application to help spectators balance opportunities to cheer for other runners and their own runner. Through a pilot study with 5 spectators and 11 runners at a 10k race, we demonstrate that 80% of spectators were able to support other runners in addition to their own, and that 64% of the runners who needed help were supported by cheers from spectators they did not know. Results show that spectators were able to support the needs of others in addition to focusing on their own goals with ease because the continual support framework helped them (a) maintain awareness of others' needs and their own goals, (b) evaluate opportune moments to support others or focus on their primary goal, and (c) prepare to provide support on-demand between their goals. Finally, our findings demonstrate that spectators valued having opportunities to support community needs between their own goals. These findings provide early evidence for how CSCW technologies can take integrated approaches to support dynamic needs in communities on-demand while being cognizant of and directly supporting the goals of potential helpers, improving and scaling how individuals support one another within these crowd and community settings.

2 BACKGROUND

Our work seeks to understand how we might enable continual support in group and community settings by deliberately helping individuals balance their own goals with supporting emerging needs of others. We build on prior research across (1) systems that facilitate help-seeking within communities; (2) systems for real-time crowd support; and (3) systems for volunteer crowd support in local communities.

2.1 Helpseeking within Communities

CSCW research has long been interested in better connecting people's needs to existing resources in their community [2]. Prior work has developed and studied systems like Expertise Recommender [25, 26] that help individuals identify experts in their organization who might be able to fulfill a help request, and flexibly filter recommendations based on social distance, expertise area, and general availability. Subsequent expertise recommendation systems follow a similar design but extend this work for identifying experts across one's social network [15, 34] and for addressing time-sensitive or location-based queries [7, 29]. Beyond recommending experts, other systems have considered how

to best connect helpers to needs in a community, for example by optimally pairing individuals who can best help one another during a scheduled help session [28]. These approaches are particularly effective in settings where reporting needs and general availability, or scheduling help sessions is sufficient for coordinating support in a community.

However, prior work does not model how a person's needs may change over time, or how the changing status of a potential helper's own goals may affect their availability to help in the moment. Without models of people's changing needs, relying on individuals to regularly report their own needs [25, 26, 28] may lead to incomplete data (e.g. needs primarily reported by those who are skilled at seeking help rather than needs across the whole community). Similarly, when a potential helper's availability depends on the changing status of their own goals, general models of availability (e.g. based on their general status [25, 26] or calendars [34]) and scheduled help sessions [28] can be unreliable predictors of who is actually available to help in the moment. Without modeling the continuously changing status of people's needs or goals, existing systems cannot know (or even support people in knowing) when someone actually has the capacity to support a need, and when they should focus on their own goals. To fill this gap, continual support models people's needs and goals explicitly, so as to enable potential helpers to maintain awareness of the moments needs arise *and* the moments goals demand attention. Modeling the status of needs and goals can be used to not only fluidly transition people between community needs and their own goals, but also better support people in achieving their own goals.

2.2 Real-time Crowd Support

Real-time crowdsourcing research has explored techniques for supporting real-time needs with paid, on-demand crowdworkers. Prior approaches suggest models to retain crowds in waiting lobbies where they can complete tasks, and readily support other needs that may arise on-demand [5]. These approaches have enabled on-demand crowds to support real-time tasks like transcribing [23] and coding video [24], captioning images [6], and monitoring video streams for specific individuals [30]. In such real-time crowdsourcing systems, workers are paid to remain generally available for completing any tasks that are presented to them.

However, in our setting, potential helpers may only be conditionally available; as volunteers, they may wish to focus on their own goals at certain times, and to help address others' needs that arise at other times. Existing real-time crowdsourcing systems that assume people are available to help and lack models of helper goals may be poorly suited for such settings where a helper's willingness to engage in support tasks is dependent on being able to attend to their own goal. In order to identify potential helpers who can actually help in the moment, and to generally support them in making effective decisions over whether to help or to focus on their own goal, continual support models people's goals explicitly and reasons about them when coordinating support opportunities. Unlike real-time crowdsourcing systems that primarily make coordination decisions based on tasking needs, continual support must additionally consider the goals and ongoing activities of those who might provide help, so that individuals can best meet their own goals and still provide support to others when they have the capacity to do so.

2.3 Volunteer Crowd Support for Needs in Local Communities

Recent work has explored approaches for engaging volunteer community members to support local community needs by providing opportunities for helping others in ways that are cognizant of a helper's routines. On-the-go crowdsourcing [14, 20] studies how people may help complete small, physical tasks for others in their community by making convenient contributions through their existing routines and routes, like helping to look for lost items or deliver a package while walking from the train en route to their workplace. Similar to our work, on-the-go crowdsourcing

systems may model the routines of potential helpers and identify opportune moments for them to provide help where their help may be most needed [21] as a way to make effective use of people's efforts and request contributions when people are likely able to help.

But unlike our work, such on-the-go crowdsourcing systems primarily focus on tasking design that is non-disruptive towards the goals of individuals (e.g. identifying convenient moments in which they can help), and are generally less focused on directly supporting the goals of helpers. In volunteer settings, a potential helper may choose to not support a community need because the task is generally inconvenient for them to do. However, when potential helpers want to support community needs between moments where their own goals require their attention, individuals may be unwilling to support needs because they struggle to assess whether or not they have the capacity to support in the moment or need to focus on their own goals. These potential helpers may need support to balance support tasks in ways that are not only non-disruptive, but directly enable them to also complete their primary goals. Thus, we may also need approaches that can consider how to prepare individuals with the detailed context and cues they need to switch between supporting community needs and re-focusing on their own goals with ease. To fill this gap, continual support primes potential helpers with the information they need to readily engage in both emerging opportunities to support the needs of others, and to re-focus on their own goals as needed.

2.4 Approaches to Crowdsourced Runner Support

There have been several recent approaches to soliciting more support for runners at races, such as systems for crowdsourcing remote support [9, 10, 31] or custom race applications designed to engage spectators at the race [3, 4, 19]. Prior work explores systems through which people remotely send "cheer" notifications to runners during their race, which runners can then receive through messages on their mobile devices [10], light indicators on companion drones [31], or haptic feedback via digital batons [9]. In recent years, major race organizations have released customized apps designed to help spectators track their own runners, and also search for and track additional runners they may want to support [3, 4, 19].

However, studies have shown that runners find remote support less impactful [10, 22], and while spectators could use existing apps to track other runners, we've observed that they still face challenges in identifying opportune moments and preparing to support other runners when they are also cheering for their own runner. Remote cheering systems can increase the number of cheers runners receive, but in studies, runners reported that individual remote cheers were only mildly motivating [10]. In practice, spectators often attempt to cheer for other runners from the sidelines during races; however, runners in our needfinding described much of this support as "zombie claps", where spectators are generically cheering for runners that go past. Runners did not find these cheers to be personally motivating, and often couldn't tell if a cheer was directed toward them or another runner. These findings are consistent with prior literature on motivating runners at races, which suggests that *directed cheers* (e.g. when a spectator shouts a runner's name) feel much more personal and motivating than generic cheers from the crowd [22]. While race organizations provide apps that allow spectators to track other runners (e.g. searching for and pinning other runners to their maps), we observed that spectators still struggle to identify opportune moments and prepare to support other runners when they are also focusing on their own runner. In contrast to these existing approaches, our work seeks to create opportunities for individuals to provide meaningful support by explicitly overcoming obstacles to identifying opportune moments and preparing to meaningfully support emerging needs (e.g. cheering for other runners) between their own goals (e.g. cheering for their own runner). By overcoming these obstacles, continual support can enable spectators to support others' needs while still being able to focus on their own goals.

3 CONTINUAL SUPPORT APPROACH

To overcome the general obstacles that individuals face in balancing supporting others' needs with focusing on their own goals, we propose ***continual support***, a conceptual approach and technical framework that (1) *monitors* the status of individual goals and emerging needs; (2) *matches* individuals to opportune moments to support others' needs given the state of their goals; and (3) *primes* individuals with the context they need to provide support on-demand (Figure 1). Continual support takes inspiration from *continual computation* [16, 18], which suggests principles and models for continually addressing problems by leveraging idle computational resources between moments when problems arise to precompute ways to address potential future challenges. We consider how to apply such principles to continually meeting needs with existing resources in community settings. To effectively coordinate such community support in real-time as needs arise, continual support models the emerging needs of individuals and the goals of potential helpers so as to optimally leverage the moments when potential helpers have capacity to support those needs on-demand. Continual support can enable individuals to support the needs of others in ways that directly enable them to attend to their own goals, and in doing so, can sustainably scale the needs that can be met with existing community resources.

In the following section, we use the practical setting of spectators cheering for their runner and other runners at races[1] to conceptualize the challenges in engaging individuals in continual support. We then describe how the ***continual support framework*** overcomes these challenges to enable individuals to balance their own goals with supporting the needs of others.

3.1 Challenges in Engaging Potential Helpers in Continual Support

3.1.1 Awareness of Moments when Needs Arise and Goals Demand Attention. While generally willing to support others' needs, potential helpers can still struggle to identify moments to support others' needs or focus on their own goals when people's needs in the community and the status of their goals continuously change. For example, a spectator may be willing to cheer on other runners at a race in addition to their own because they are familiar with the positive impact of a directed cheer from prior race experience. However, a spectator may struggle to track and directly cheer for their other runners at the same time as their own. Without supports for detecting specific moments to support other runners or focus on their runner, we observed that spectators generally provided support that was less impactful, selective, or diverted attention from their own goals. For example, we observed that most spectators would passively clap or generically cheer things such as "let's go runners!" along the sidelines rather than directly cheer. Some spectators would try to directly cheer for specific individuals, but often defaulted to selectively supporting the needs of those who best solicit help (e.g. runners in costume). In other cases, we observed that some spectators would expend extra effort to gather information about runners passing by in ways that might momentarily distract them from keeping track of their own runner (e.g. scanning the crowd and shouting out names on runners' bibs). Without systems that model the changing moments where needs arise and goals demand attention, potential helpers can struggle to maintain awareness of changing needs across a community while still being cognizant of their own goals.

3.1.2 Deciding When Support Others' Needs or Focus on Goals. Despite having windows of availability to support others' needs, potential helpers can struggle to decide whether or not they have the capacity to support a need when their moment-to-moment availability depends on the changing status of their own goals. For example, while a spectator may have significant

[1]Over the course of 4 years, we took a design-based research approach [11, 33] to conduct 5 major iterative cycles of needfinding, designing, and testing in live race settings to understand the challenges of engaging the ad-hoc crowd of spectators in continual support.

idle time at races to cheer for other runners, the moments when their own runner approaches may coincide with emerging opportunities to cheer for nearby runners, making it challenging to determine whether they have the capacity to support another runner in the moment. Without ways to evaluate whether or not they had the capacity to provide support, we observed that spectators often refrained from supporting the needs of others. For example, some spectators who tested earlier prototypes described how they would hesitate to support others in the moment (e.g. cheer for other runners) if it was not explicitly clear to them that their primary goal did not require their attention in the moment (e.g. their runner was still far away). Without systems that consider the goals of potential helpers to determine their moment-to-moment availability, helpers can struggle to evaluate when they actually have the capacity to support others' needs in the community.

3.1.3 Preparing to Support Others Needs On Demand Between Goals. While generally well-equipped with the skills or prior experience to offer meaningful help, potential helpers can still struggle to provide impactful support on demand between the moments when their own goals demand their attention. While a helper may know about a support opportunity and be generally well-equipped to provide that support, without the necessary context and cues required to give that support on-demand, the support interaction can break down. For example, the strategies that we observed spectators use would often require coordinating in advance with their runners to get pace charts for estimating arrival times at course locations based on the runner's expected pace, or knowing in advance what outfit they would wear on race day so they could easily spot them in the crowd. Similarly, spectators who attempted to connect and engage with specific runners in impactful ways would expend effort scanning the crowds for context to help them cheer, like a runner's name on their bib, or jerseys that represented a country or favorite team, which they often did not see until the runner was already racing past. Without systems that provide the context and cues that reduce the inherent friction of engaging in an opportunity to support another's needs, and smoothly transitioning focus back in the moments where goals demand attention, potential helpers can be unprepared to support others' needs on demand between their own goals.

3.2 Continual Support Framework

To overcome challenges in engaging potential helpers in continual support, the ***continual support framework*** (Figure 2) *monitors* the changing status of people's goals and emerging needs; *matches* individuals to opportune moments to support others' needs or focus on their goals; and *primes* individuals with the context they need to provide support on-demand. The framework then *verifies* support interactions and *updates* the state of needs and resources across the community. We detail these framework components below.

3.2.1 Monitor: Tracking the Changing Status of Needs and Goals. Monitor enables a system to determine when needs arise and goals require attention by modeling people's needs for help and helpers' own goals (see Figure 3). We expect that modeling and maintaining a continuous understanding of changing needs and goals will allow the system to surface moments to support the needs of others and focus on goals, offloading the monitoring effort from potential helpers. To model help-seekers and their changing needs, Monitor implements a *help-seeker monitor* and a *needs trigger*. To model helpers and their changing goals, Monitor implements a *helper monitor* and a *goal trigger*. We detail the components of Monitor below.

(a) The ***help-seeker monitor*** models the factors that signal an individual's changing progress and possible obstacles to determine the conditions where a person may need help (see Figure 3A). The help-seeker monitor can encode and model common heuristics that signal a person's ongoing progress and obstacles (e.g. encoding the changing pace or biometrics of a

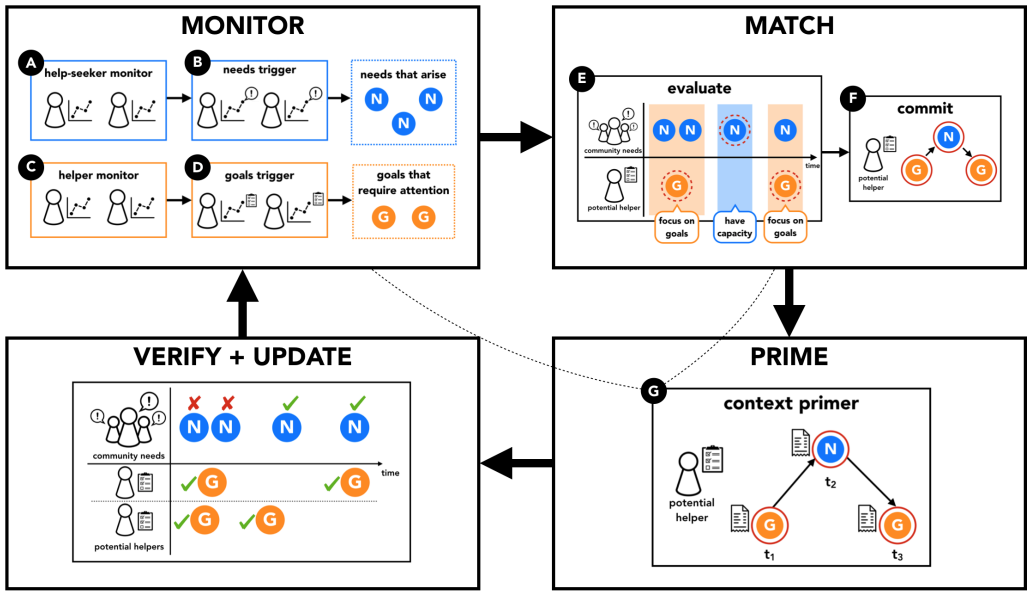


Fig. 2. The continual support framework (a) *monitors* the changing status of people's goals and emerging needs; (b) *matches* individuals to opportune moments to support others' needs or focus on their goals; and (c) *primes* individuals with the context they need to provide support on-demand. The framework then (d) *verifies* support interactions and *updates* the state of needs and resources across the community.

runner to determine their ongoing status and signal pain points during a race, or a student's ongoing progress on their study design to signal risks in their research argumentation or project timeline). The help-seeker monitor can also model patterns of a help-seeker's needs over time as a way to track their skills, growth, and development in the community (e.g. tracking race segments where a runner commonly requires the most support, or recurring patterns of challenges a student faces in their research). Finally, the help-seeker monitor can model the attributes of needs themselves. For example, the monitor can track the specific expertise or support required to resolve a need (e.g. a specific research skill, or a specific cheer that a runner wants to hear); or the priority of needs per individual and across the community (e.g. the impact of not resolving a need in the moment it arises, whether a need has been resolved, or which individuals have more unmet needs).

(b) The ***needs trigger*** then continually assesses the factors modeled in the help-seeker monitor to detect specific moments when a help-seeker may have a need (see Figure 3B). To detect needs in the moment, the needs trigger listens to the help-seeker monitor and identifies when an individual may face an obstacle or require support to signal that a need has emerged (e.g. detecting an event where a runner's pace drops off, or a student starts work on study protocol before getting feedback on measures).

(c) The ***helper monitor*** models the factors that signal the changing status of a person's goals to determine when they might have windows of capacity to support needs in the community (see Figure 3C). For example, a helper monitor may encode the progress of a spectator's own runner to determine whether they have the capacity to cheer for another runner in the moment, or encode whether a student is making their desired progress, meeting deliverables, or hitting milestones in their planned research agenda to determine whether they currently

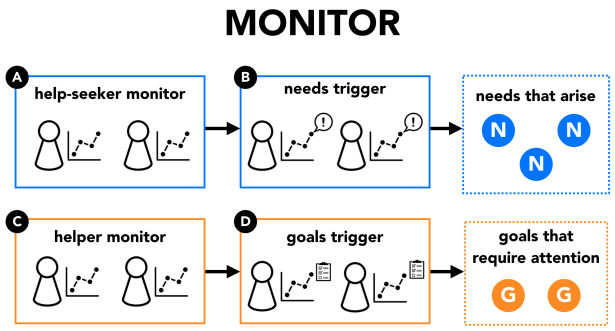


Fig. 3. The ***Monitor*** component of the continual support framework enables a system to determine when needs arise and goals require attention by modeling people's needs for help and helpers' own goals.

have the capacity to help a peer with their study design. When helpers have multiple goals (e.g. a spectator has a friend and family member in the race, or a graduate student is working on a manuscript and a conference talk), the helper monitor can also model the priorities across different goals. Finally, the helper monitor models factors that inform a potential helper's changing ability and willingness to help. For example, it may model a helper's expertise or prior experience (e.g. whether a student has worked through a study design previously, or if a spectator has cheered for other runners in the past); the convenience of a support task (e.g. tracking which other runners are close by to a spectator and therefore convenient for them to track and cheer, or if a helper is working on a similar type of task, like writing up their own findings and primed to think about study design); or support contributions over time to inform how the system might coordinate and distribute support across the community.

(d) The ***goals trigger*** then continually assesses the factors modeled in the helper monitor to detect the specific moments where a potential helper's goals require attention (see Figure 3 D). To detect when a potential helper needs to focus on their own goal and does not have the capacity to support others, the goals trigger listens to the helper monitor and signals when an individual's goals may require attention in the moment (e.g. an event like a missed project milestone or when a spectator's runner is approaching).

3.2.2 Match: Identifying Opportune Moments to Support Needs or Focus on Goals. Match enables a system to match potential helpers to opportune moments to support others' needs or focus on their own goals by evaluating when potential helpers have the capacity to support an emerging need in the moment (see Figure 4). We expect that computing, for each potential helper, the needs they can support and the goals that may demand their attention will sustainably engage such helpers by helping them determine when they actually have the capacity to support an emerging need, and when to focus back to their own goals in the moments when those goals demand their attention. To enable helpers to evaluate their capacity to support, Match implements *evaluate*, and *commit*, detailed below.

(e) The ***evaluate*** process continually reviews the moments that needs arise and goals require attention detected by Monitor to compute, for each potential helper, the opportune moments to support the needs of others or focus on their own goals (see Figure 4E). Generally, evaluate can determine when a person has capacity to support another's need by assessing whether an emergent community need is in conflict with a moment where the helper must focus on their goals (e.g. a spectator may not have capacity to support another runner when their

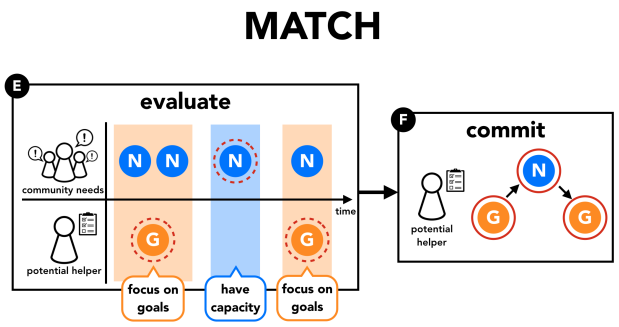


Fig. 4. The ***Match*** component of the continual support framework enables a system to match potential helpers to opportune moments to support others' needs or focus on their own goals by evaluating when potential helpers have the capacity to support an emerging need in the moment.

own runner approaches at the same time, or a student may have capacity to offer study design feedback to a peer after reaching their own paper deadline). The system can then use this information to differentially emphasize opportune moments to support others' needs or deliberately shift a helper's focus back to their own goals in the interface. When optimally matching a helper to specific opportune moments to support a need, evaluate consider factors modeled by Monitor, such as the priority of goals or needs, the convenience of a support task, the expertise of a person for supporting a need. Finally, evaluate computes, for each help-seeker, the potential opportunities to receive support for a given need (e.g. when spectator may potentially cheer for a runner, which could be shared with the runner or used to throttle the frequency of runner updates to ensure well-timed support interactions; or when there are multiple support resources that can potentially meet a student's needs, which they could then use to make decisions about how to leverage different resources in the community).

(f) The ***commit*** process logs a potential helper's commitment in the moment they decide to either help with a need or to focus on their own goal (see Figure 4F). These commitments can later be used to determine whether a helper is available to be matched to another need, or is currently engaged in an activity. For example, commit could log when a spectator chooses to cheer for for another runner, which would allow the system to know they are committed for the time being and focus on preparing them to cheer for the specific runner. Alternatively, commit could also log when a graduate student has committed to their own upcoming paper deadline for the time being and focus on preparing them to execute their writing tasks.

By evaluating both the set of possible needs and the ways in which their goals may require their attention in the moment, continual support systems can distill for potential helpers the information they need to decide whether to focus on their goals or support a community need in the moment. Rather than notifying potential helpers of every support opportunity, a continual support system can use its understanding of both emerging needs and the status of goals to strategically draw a potential helper's attention towards emerging needs when goals do not demand attention, or direct the helper's attention towards their goals when they do.

3.2.3 Prime: Preparing to Support Needs On Demand Between Goals. Prime gathers detailed context based on static and dynamic information needed to provide on-demand support or smoothly transition back to goals, and uses this information to prime and cue individuals once they've committed to either a support opportunity or focusing on their goals (see Figure 5). We expect that

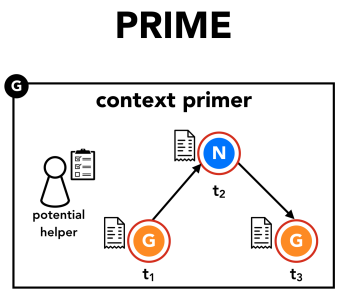


Fig. 5. The ***Prime*** component of the continual support framework gathers detailed context based on static and dynamic information needed to provide on-demand support or smoothly transition back to goals, and uses this information to prime and cue individuals once they've committed to either a support opportunity or focusing on their goals.

priming and cueing potential helpers with this detailed context will prepare them to both offer support in the moments between their own goals, and reorient towards their own goals in ways that deliberately support the goals of potential helpers. To gather and deliver context to potential helpers, Prime implements a *context primer*, detailed below.

(g) The ***context primer*** gathers static and dynamic information from Monitor and Match after the helper has committed to either a need or their goal, which are then used to prime and cue potential helpers for specific tasks (see Figure 2G and Figure 5G). Prime gathers static information about the needs themselves that might prepare helpers to provide impactful support (e.g. pictures of runners and their personalized cheer, or project descriptions from a student). Prime also gathers dynamically changing information that might prepare a potential helper to support on-demand (e.g. the changing location of a runner as they approach the spectator, or a timeboxing timer that helps a graduate student focus the limited feedback time they have with the peer they are helping). The system can then use this information to strategically prime and cue individuals with the information they need to fluidly move between effectively supporting needs and focusing on their goals.

Taking inspiration from microtasking research, which explores the challenges of providing context to individuals as they task switch [8, 27**?**], we explore how such priming and cueing of support interactions is particularly important in these settings where helpers may need to return to their own goals shortly after committing to help another person. By providing potential helpers with the detailed static and dynamic information needed to effectively support another's need or return focus to their own goal, continual support systems attempt to alleviate some of the inherent friction in supporting a need on the fly.

3.2.4 Verify and Update: Tracing Interactions across Support Ecosystem. To enable ways of tracking support interactions and the updated state of needs and resources across the community's support ecosystem, we include *Verify* and *Update* components in our framework (see Figure 2). The ***Verify*** component seeks information about potentially completed support tasks through three processes: *deliver*, which listens for information about the support provided/attempted by the helper; *receive*, which listens for whether the help-seekers received any support for their need; and *response*, which detects whether that need was met. The ***Update*** component periodically reads from the processes in Verify to update the system's understanding of the overall support ecology – the state of changing needs and the availability and contributions of volunteer helpers across a

community. The Update component takes the needs that are now completed (or remain outstanding) and potential helpers that may now have capacity (or not have capacity), and writes back to the help-seeker and helper monitors in Monitor that maintain such information. By keeping track of the outcomes of individual support interactions and the general understanding of needs and the capacity of potential helpers in the support ecosystem, continual support can trace each support interaction end-to-end, and further, can capture and model ecosystem level representations of support in the community. In larger scale settings, such support ecology representations can be used to design algorithms that coordinate support across the community in particular ways.

3.3 Framework Implementation Details

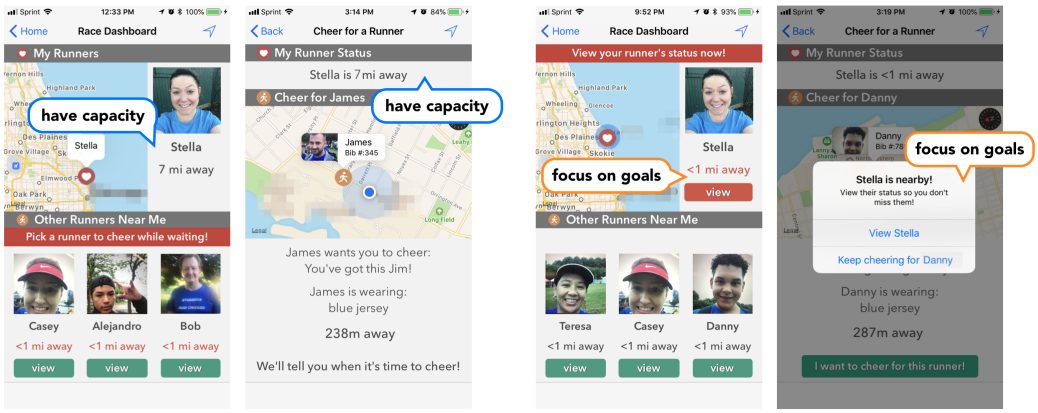
The continual support framework is implemented with a series of protocols that represent the core architectural components detailed above (see Figure 2). We deliberately focus on a protocol-oriented implementation as a way to forefront the need to continually model the evolving state of emerging needs of help-seekers and goals of potential helpers in the community when coordinating support. To maintain up-to-date information on help-seekers, potential helpers, and their needs and goals, it was critical for our implementation to track their continuously changing status, without compromising system performance. Monitor periodically logs updates with low frequency, and then strategically throttles the frequency of updates as needed. For example, Match evaluates for both help-seekers and helpers whether there is a potential helper with the capacity to support an emerging need. In response, Monitor increases the frequency of updates just before the potential helper will have capacity, and further increases update frequency in the moments where the helper does have the capacity to support the need. Once the opportune moment has passed, Monitor then reduces update frequency back to the baseline. To accurately present real-time updates and represent opportune moments to helpers despite real-world latency issues, Monitor calculates perceived delays to estimate a person's status based on their last logged update (e.g. factoring in network delays to estimate a runner's current location based on their last known location; the time since their last location update; and the runner's pace, heading, and bearing, or surfacing to a mentor that a student has not updated their study design in over two weeks).

4 CROWDCHEER: CONTINUAL SUPPORT TO ENGAGE AD-HOC CROWDS

We study the continual support framework in the context of ***ad-hoc crowds*** – groups of individuals who may not have explicit social ties as they would in a work or learning community, but because they are engaged in a similar activity in a shared setting, they may be generally willing, available, and able to support those who face similar challenges. To illustrate how continual support can enable individuals to balance their own goals with supporting others' needs, we focused on the practical setting of an *ad-hoc crowd* of spectators at races. We designed and implemented ***CrowdCheer*** (Figure 6), a mobile application that connects spectators to opportunities to cheer for other runners at a race while still cheering for their own runner. Implementing the continual support framework, CrowdCheer (1) monitors the changing state of the spectator's runner alongside emerging opportunities to cheer for nearby runners; (2) matches spectators to opportune moments to support other runners or focus on their own runner; and (3) primes spectators with the context they need to cheer on either their runner or another runner on-demand.

4.1 Monitoring Needs and Goals in CrowdCheer

Implementing Monitor, CrowdCheer uses a *dashboard* to continuously present the status of the spectator's primary runner alongside nearby runners who may need support (see Figure 6a (Left)). The top half of the dashboard presents the continuously changing status of the spectator's primary runner using a live map and a distance tracker, while the bottom half of the dashboard continuously



(a) emphasis on others' needs (b) emphasis on own goals

Fig. 6. **CrowdCheer** is a mobile application that implements the **continual support framework** to enable spectators to cheer for their runner and other runners at a race. CrowdCheer's **dashboard** displays the continuously changing status of their runner alongside other nearby runners who may need support. CrowdCheer uses **sideloading** to emphasize other runners when their own runner is far away (a) and the spectator's own runner when they are nearby (b).

updates with a list of other nearby runners who the spectator can support. We hypothesize that a dashboard that continuously displays the changing status of a potential helper's goals, and the emerging needs of others will help the ad-hoc crowd of spectators easily maintain awareness of moments where different needs arise and their goals require attention.

In order to display this information in the dashboard, the continual support framework *monitors* the changing status of spectators and their goals (e.g. the status of their runner) and runners and their emerging needs (e.g. when a runner could use a cheer). To model spectators and their goals, the *helper monitor* tracks the spectator's changing location and the bib number of their primary runner(s), and the *goals trigger* recognizes when the spectator's runner is within cheering range. To model runners and their needs, the *help-seeker monitor* tracks the changing location, pace, and profile information of each runner (e.g. their name, picture, outfit, bib number, their preferred cheer, etc.). The *needs trigger* assumes that runners could use support throughout the race for simplicity, and is used to detect when a potential helper may be nearby the runner as a way to throttle the frequency of location updates.

4.2 Matching to Opportune Moments in CrowdCheer

Implementing Match, CrowdCheer evaluates a spectator's moment-to-moment capacity to support others' needs and uses a *sideloading* technique to conditionally emphasize opportune moments to cheer for other runners or their own runner (see Figure 6). To illustrate sideloading, when a spectator has the capacity to support others (e.g. their own runner is far away, see Figure 6a), CrowdCheer de-emphasizes the top half of the Dashboard displaying the status of their primary runner (e.g., using grey text and colors), and emphasizes the bottom half of the Dashboard displaying opportunities to cheer for other runners nearby (e.g., using red text and colors), see Figure 6a (Left). If the spectator is not actively looking at the dashboard, CrowdCheer periodically notifies the spectator about opportune moments to support others with a message that reads: "Your runner is 5 mi out and is doing well, but some nearby runners could use your help while you wait!" To

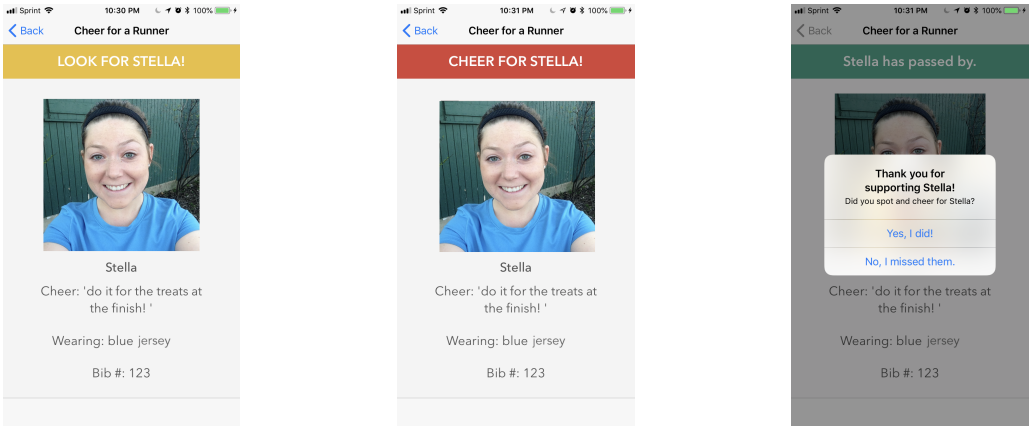


Fig. 7. Implementing Prime, the **priming cues** in CrowdCheer provide identifying information, support information, and visual cues for spectators so that they know precisely when to look up and search for the runner in the crowd (left), when and what to cheer (center), and verify if they were able to cheer for the runner once they've run past (right).

ensure that spectators still have the capacity to cheer in the middle of a commitment (e.g. that their runners are still far away when they are actively cheering for other runners), CrowdCheer continues to monitor and display their runner's status above in the cheering view ("Stella is 7mi away"), see Figure 6a (Right). When a spectator's own runner is nearby (Figure 6b), the Dashboard now emphasizes the status of their own runner in the top half of the Dashboard and de-emphasizes other nearby runners' needs for help (see Figure 6b, Left). Should a spectator's own runner approach them while the spectator is not actively looking at the dashboard or is actively preparing to cheer for another runner (Figure 6b, Right), a notification is sent to alert them that their runner is nearby with the option to view their runner's status. We hypothesize that sideloading to conditionally emphasize needs and goals to a potential helper can guide the ad-hoc crowd of spectators as they make decisions about when to focus on their own goals, and when they have the capacity to support others' needs.

In order to match spectators to these opportune moments, the continual support framework continuously *evaluates*, for each spectator, the moments where needs arise (e.g. when there is a nearby runner) and the moments where goals require attention (e.g. when their runner is approaching) and computes moments to support others' needs or focus on goals. CrowdCheer then uses this information to inform how the sideloading technique conditionally emphasizes opportunities to spectators in the interface. Once a spectator chooses to cheer for either their runner or another runner, *commit* logs this decision, which it can then use to inform how a potential helper's moment-to-moment capacity changes (e.g if they've committed to cheering for another runner, we no longer present opportune moments to support other nearby runners for the time being).

4.3 Priming for Opportune Moments in CrowdCheer

Implementing Prime, CrowdCheer provides *priming cues* (i.e. identifying information, support information, live maps, distance trackers, and visual cues) to prepare spectators to successfully a cheer on-demand once they commit to either their runner or another runner. The cheering view (see Figure6a, Right) initially provides spectators with *identifying information* consisting of a picture of the runner, their bib number, and their race outfit description. This identifying information is

displayed alongside *support information* that show the runner's name and a personally meaningful cheer that the runner had requested through the app. Also displayed are a *live map and distance tracker* that show the approaching runner and their distance from the spectator, which shifts to red text as the runner approaches. As the runner gets within 100 meters of the spectator and location tracking accuracy degrades in usefulness, the cheering view switches to a simplified view (see Figure 7) that uses a series of *visual cues* that provide color indicators to let spectators know when to look up (in yellow, Figure 7, left) and when to cheer (in red, Figure 7, center). Once the runner has passed the spectator (in green, Figure 7, right), the system prompts the spectator to verify whether the spectator has successfully supported the runner. We hypothesize that providing potential helpers with static context (i.e. identifying and support information) and dynamic context (i.e. live maps, distance trackers, and visual cues) will structure and guide the support interaction in a way that prepares individuals to support others' needs, or seamlessly return their focus to their own goals in the moment.

In order to prime and cue spectators effectively, the *context primer* within the framework's *prime* component retrieves the necessary static context about either the spectator's goal or another's need (e.g. their runner or another runner), and dynamic information about the changing status modeled in the helper and help-seeker monitors. Once a support opportunity ends (i.e., the runner has passed the spectator), *verify* listens for confirmation and signals whether or not the spectator successfully supported the runner, and passes this information to *update*, which logs who recently cheered or received cheers. The information about who recently cheered is then used to implement a short delay in sending local notifications to spectators who may have just supported another runner.

4.4 CrowdCheer Implementation Details

CrowdCheer is implemented as an iOS mobile application in Swift 3 for iOS, that uses a MongoDB and Parse Server backend. The iOS app implements the interface and system architecture models, and the Parse Server, hosted on Heroku, manages access to the database, which maintains tables of spectators, runners, the historical and current locations of each, and the global set of support opportunities that spectators committed to and engaged in.

5 CONTINUAL SUPPORT SYSTEMS PILOT STUDY

To study how the continual support framework might enable an ad-hoc crowd of spectators to balance their own goals with supporting the needs of others, we conducted a pilot study with 11 runners and 5 spectators at a 10K race hosted in a major U.S. city. Runners and spectators used CrowdCheer at the race and completed post-surveys and semi-structured interviews soon after. We hypothesize that *continual support systems* will enable potential helpers to balance supporting needs of others with focusing on their own goals in the following ways:

- **H1:** potential helpers will maintain awareness of the moments when their goals require attention and different needs arise across the community;
- **H2:** potential helpers will make effective decisions about when to focus on their own goals or support others' needs in the moment; and
- **H3:** potential helpers will be prepared to support others' needs or return their focus to their own goals on-demand.

5.1 Recruitment Method

To evaluate the potential of continual support systems for engaging ad-hoc crowds, we strategically recruited spectators that would struggle to balance supporting others (e.g. cheering for other

runners) with focusing on their own goal (e.g. cheering for their own runner). We coordinated with race organizers to set up a study recruitment table at the packet pickup event the day prior to the race. Interested runners and their spectators were informed that the app would help them track and cheer for their runner, and that the spectator would also see opportunities in the app to support other runners if they so desired (but that this wouldn't affect compensation). While any interested runner could sign up, we selected for spectators who were there to support a particular runner, and who confirmed that they typically cheer for other runners at races while they wait. Runners who signed up at packet pickup without a spectator were instructed to recruit their spectators and were provided with a standard recruitment message. 11 runners and 5 spectators participated in the study. Runners and spectators were compensated with $30 for participating at the race and completing post-surveys and interviews, and runners received an additional $10 for every spectator they recruited (up to 5).

5.2 Study Procedure

In the study procedure, runners were asked to create profiles and track their race using CrowdCheer, and spectators were asked to use CrowdCheer as it was useful for cheering during the race. Runners and spectators who signed up for the study at packet pickup installed CrowdCheer on site. They were provided with instructions on how to use the app on race day, and one of the authors verified that they understood how to set-up and use the app for the race. Runners were instructed to enter identifiable information and personalized cheer requests that spectators use to cheer for them, and to carry their phone with them during the race so that CrowdCheer can track their run in the background. Spectators were instructed to be out on the race course as they would, and to use CrowdCheer only as it was useful to cheer for runners at the race (i.e., they are not expected to be actively using CrowdCheer at all times).

5.3 Data Collection

To collect data on how spectators and runners used CrowdCheer to provide and receive support at the race, participants completed post-surveys and video/phone interviews following the race. Spectators and runners were asked to complete 15 minute post-surveys within 12 hours of the race. A subset of participants (5 spectators and 3 runners) were asked to complete 20-30 minute semi-structured video/phone interviews within a week of the race[2].

To evaluate which runners each spectator attempted to support at the race, and collect detail about each support interaction, we asked spectators and runners to recall details about the cheers they gave and received at the race. Due to data gaps in our interaction logs[3], we confirmed cheering interactions by using a stimulated retrospective think aloud (RTA) method [13] in our post-surveys

[2]In our runner post-survey instrument, runners were asked to recall times that spectators specifically cheered their name, or the personalized cheer they had entered in the app. Many of the runners remembered hearing directed cheers from the crowd. However, given that cheers occurred in fleeting moments, most runners struggled to provide enough detail to confirm which of the spectators specifically cheered for them. As a result, we decided to omit summative data collected from runners in our analysis.

[3]While we collected data logs of cheering interactions, we found gaps in our data after surveying and interviewing our participants. Many of our spectators detailed specific cheering interactions that did not have corresponding data entries in our interaction logs. We suspect that one of two issues may have occurred. It's possible that due to the network latency issues in the race setting, and the high traffic of storing and streaming location updates with each spectator, that some data from interaction logs was lost. Another possible explanation is that once spectators were cued to look up and scan the crowd for a runner (see Figure7, Center), that they missed the momentary assessment to confirm their cheering interactions after the runner ran past. To mitigate the risk of inaccurate self-report, we utilized a stimulated retrospective think aloud (RTA) method [13] in our post-surveys and interviews, and conducted them within 12 hours and a week of the race, respectively to confirm cheering interactions.

and interviews. In our spectator post-survey instrument, spectators were asked to review each possible runner in the system and recall details about which runners they cheered for during the race. Runner profiles (picture, name, bib number, outfit description, custom cheer) were provided to help stimulate spectators' memories. For each runner, we asked spectators to select what they recalled about cheering for that runner from the following options: (a) This was my runner, (b) I selected I want to cheer for this [other] runner, (c) I spotted this runner on the course, (d) I cheered for this runner, (e) The runner saw/reacted to me cheering, and (f) I do not recall this runner. Through a series of open-ended questions, we asked spectators to narrate highlights of that experience, and recount specific instances and details of cheering for other runners. In our interviews, we asked spectators which runners they recalled cheering for by showing them runner profiles again to cross-verify with the data they reported in the post-surveys, and asked them to narrate details about each cheering interaction they reported.

5.4 Data Analyses

To evaluate the ease, disruption costs, and overall experience of trying to balance supporting the needs of others with focusing on their primary goal, spectators were asked whether the app was disruptive to their primary goal of cheering on their own runner. In our interviews with spectators, one of the authors walked through each of these specific cheering instances and, situated in these concrete examples, asked them what it was like to cheer for another runner when they were also trying to track their own runner, what was it like to be part of supporting another runner's race, and why they chose to support other's needs in their idle time at the race.

To assess how continual support helped spectators identify moments where different needs arose and their goals required attention, spectators were asked if they always knew where there runner was through the app, and asked how they used the app to learn about both their runner's changing status and the emerging needs of others.

To assess how continual support helped spectators make effective decisions about when to support others' needs or focus on their own goals in the moment, spectators were asked to narrate specific stories of the decisions they made. Specifically, spectators were asked to (a) recount times they cheered for other runners and in those moments, how they knew when to cheer for other runners over their runner, (b) recount times they cheered for their runner over other runners and how they knew to cheer for their own runner in the moment, and (c) recount other times they chose not to cheer for other runners, and the reasons that may have led to those decisions.

To assess how continual support helped spectators prepare to support the others' needs or return their focus to their own goals on demand, spectators were asked about which context might have prepared them to cheer for their runner and other runners. Specifically, spectators were asked (a) if they had enough information to spot and cheer for runners, (b) what they recalled seeing in the app as they cheered, and (c) which features were most helpful for spotting and cheering for particular runners on demand.

6 PILOT STUDY FINDINGS

Below we present findings from an initial pilot study which demonstrate that the ad-hoc crowd of spectators were able to use a continual support system, CrowdCheer, to balance supporting others' needs with focusing on their own goals with ease. We then detail evidence which demonstrates how the core characteristics of the continual support framework, as implemented by CrowdCheer, enabled potential helpers maintain awareness of, evaluate, and prepare for the moments where goals demand attention and needs arise. Finally, we present findings that illustrate how ad-hoc crowds valued and enjoyed the ways in which the system created opportunities to balance community needs with their own goals. We detail these findings below.

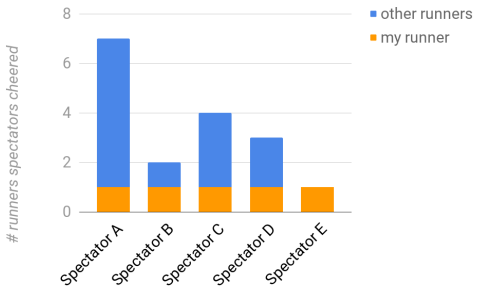


Fig. 8. All 5 spectators used CrowdCheer to successfully cheer for their own runner, and 4 of the 5 spectators recalled using CrowdCheer to collectively cheer for 12 runners that were not their own.

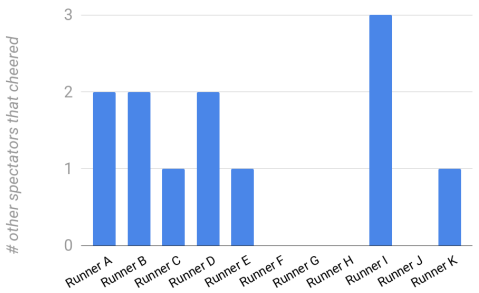


Fig. 9. Spectators recalled cheering for 63.4% of other runners using CrowdCheer.

6.1 Individuals balanced supporting others' needs with focusing on their own goal

We found that individuals were able to balance supporting others' needs with focusing on their own goal while using CrowdCheer. Specifically, results show that all 5 spectators used CrowdCheer to successfully cheer for their own runner, and 4 of the 5 spectators recalled using CrowdCheer to cheer for runners that were not their own. Spectator E, who did not cheer for other runners, was recruited by her own runner outside of packet pick-up times and described in her interview how she did not understand that the app could be used to cheer for other runners nearby beyond their own (who she did cheer for). Collectively, spectators recalled cheering for 12 runners that were not their primary runner (Figure 8). This resulted in 7 of the 11 runners in our pilot study receiving support from one or more spectators other than their own; Figure 9 shows the number of other spectators that directly cheered for each runner. These findings suggest that continual support systems which support individuals in balancing both their own goals with supporting the needs of others in their community may also help increase the coverage of needs that can be met with existing help resources.

6.1.1 Individuals felt that balancing supporting others' needs with their own goals was easy. Overall, individuals found it easy to support both the needs of others while still being able to focus on their own goals while using CrowdCheer. For example, Spectator D said in her interview: *"Not extra work to cheer for other runners, it was like BAM! Felt like a scoreboard with a starting player."* In her interview, Spectator B talked about how cheering for others was actually easier than cheering for her own runner, given that she had an additional task of helping her nephew spot his parents when cheering for her own runner: *"I think they were both pretty easy, and it was probably easier actually in my case to cheer for other runners as opposed to my runner, because you know, helping my nephew and showing him "oh do you see them? Do you see mommy and daddy?" so it was almost easier to cheer for other runners because I didn't have to worry about whether or not he could see someone in particular."* In this example, the spectator described the ease of cheering for both her primary runner and other runners, despite having an additional point of focus (i.e. helping her nephew) not modeled in the system. These findings suggest that by helping individuals balance both supporting others' needs with their own goals, continual support systems can alleviate some of the friction that may have previously prevented individuals from supporting others in impactful ways.

6.1.2 Individuals felt supporting others' needs was non-disruptive to their primary goals. Individuals found that they were able to support the needs of others in a way that was not disruptive to their primary goals when using CrowdCheer. Specifically, when asked if they felt the app was non-disruptive about opportunities when they were focused on their own runner, spectators responded with an average of 6.8 out of 7 (between agree and strongly agree, SD = 0.44). In her post-survey, Spectator D wrote that this was because "[CrowdCheer] didn't feel forceful or constantly alerting me." In her interview, Spectator A said that CrowdCheer was not only not disruptive, but actually, helped her to direct her attention back to her own runner when her runner was approaching: *"Not at all! In fact I clicked on one of the [other] names just to see how it worked and the app was like '[name of her own runner] is less than a mile away from you, do you still want to continue to cheer for this person?' and I was like oh that's pretty cool! I loved the fact that it kept giving you these 'okay, your person is coming close, do you want to keep cheering, or...?"'* These findings suggest that continual support systems are capable of enabling volunteer helpers to provide support without taking their focus away from their primary goals, and even prompting them to refocus on their primary goals as they demand attention. By framing support requests in ways that are cognizant of the individual goals of volunteer helpers, continual support systems can engage individuals like those in the ad-hoc crowd to support more emerging needs of others, while simultaneously enabling them to attend to their goals as needed.

6.2 Individuals maintained awareness of, evaluated, and prepared for moments when goals demand attention or needs arise

6.2.1 Individuals maintained awareness of their goal and others' needs. Spectators using CrowdCheer described how the Dashboard enabled them to maintain awareness of the changing status of their own goals alongside the emerging needs of others. Specifically, spectators were aware of the continuously changing status of their own runner. When asked if they knew where their runner was at all times, spectators generally agreed with a 6.2 of 7 (between agree and strongly agree, SD = 0.83) on average. Spectator C wrote that the CrowdCheer app was "giving me their distance at all times" and Spectator A wrote "Perfectly!! It was spot on with the tracking and the distance my runner was away from me." Spectators were also aware of emerging opportunities to cheer for other runners, demonstrated by the 4 of 5 spectators who cheered for other runners during the race. The one exception was Spectator E, who as mentioned previously did not fully understand the feature. When asked in their post surveys how the system helped them keep track of other runners, Spectator A wrote "It showed me the people who were around and gave me the option to cheer for them as well" and Spectator B wrote "by showing me who was near me at any given time." Further, spectators described how the Dashboard helped them simultaneously monitor for opportunities to cheer for others alongside the status of her runner with ease. In her interview, Spectator B said *"It was super easy. If I remember correctly, the top half had my runner and where she was and then the bottom half that had other runners and so you know roughly how far away they were, and you could tap on them and see more. So I mean it was just super easy to use."* These findings suggest that continual support systems can offload the monitoring effort of potential helpers in ways that enable them to maintain awareness of both the changing status of goals and emerging needs across the community.

6.2.2 Individuals evaluated when to support others' needs vs. focus on their own goals. Spectators using CrowdCheer described how sideloading helped them decide when they had capacity to support others' needs in the moments where their primary goals did not demand their focus. We found that when spectators did choose to cheer for other runners at the race, a key contributing factor was that they could see their runner was far away. When asked if they knew their runner

was not nearby when they chose to cheer for other runners, spectators responded with a 6.4 out of 7 (between agree and strongly agree, SD = 1.34) on average. In interviews, spectators referred to specific sideloading affordances that helped direct their attention towards opportunities to cheer on other runners. For example, Spectator B said: *"If I was pulling out my phone to see where they were, I could see, you know, they [their runner] were like 2 miles, a mile and a half out or whatever, and so I knew that I didn't have to worry about missing them coming through, so that gave me the opportunity to see other runners that were nearby and cheer for them."* These findings suggest that continual support systems can evaluate for opportune moments to support others and explicitly emphasize them in ways that direct individuals to focus on support opportunities in the moments they have the capacity to help.

Similarly, spectators using CrowdCheer described how sideloading helped them decide when they did not have capacity to support others' needs and needed to focus on their own goals. Spectators knew not to cheer for other runners when they could see that their own runner was approaching. When asked in the post-survey how they knew to return their focus to their own runner in the moment, 4 of 5 spectators (excluding Spectator E) described ways in which the sideloading technique informed them through different forms of notifications or dashboard information. For example, Spectator A wrote "The app said your runner is near do you still want to cheer for this person."; Spectator B wrote that they "saw the notification"; Spectator C wrote "the app alerted me" and Spectator D wrote "It informed me my runner was next or approaching". Some spectators described how at times they would ignore information about other runners because they knew to focus on their primary goal. In her interview, Spectator A said: *"So when [her runner] started getting close, I wasn't paying as much attention to other runners. Like if I saw them, and there was a couple faces that I recognize or remembered from the screen – because I kept flipping back and forth between those profiles – [her runner's name] and someone else's — so I kind of had their face in my head, so you know I didn't completely ignore them, because I knew they would be finishing around the same time, but I didn't pay as much attention when I saw when he [her runner] was coming up."* In this example, despite having access to information and a choice to support other runners, the spectator was able to primarily focus on her own runner, while maintaining peripheral awareness of other opportunities that were out of focus in the moment. These findings suggest that continual support systems can evaluate for opportune moments to focus on goals and explicitly emphasize them in ways that help individuals shift their focus away from supporting needs in the moments where their primary goals require their attention.

We also sought to understand situations where individuals chose not to support others' needs, despite CrowdCheer identifying an opportune moment to do so. For example, Spectators B and D also described moments early in the race where they saw opportunities to support others but decided not to cheer because they were still orienting to the race and to using the app. We also found that spectators sometimes had other responsibilities to attend to. In her interview, Spectator B described how watching her nephew was an additional task that preoccupied her and was often a higher priority than cheering for others. While these factors were not modeled in the continual support framework, they are consistent with our finding that individuals may only be willing to provide support with they are not preoccupied with their own goals. These findings begin to suggest opportunities for future work to consider how to capture and represent additional goals that may also influence a person's capacity to provide support in the moment.

6.2.3 Individuals were prepared to support others' needs on-demand between their goals. Spectators using CrowdCheer described how the static and dynamic context in the cheering view prepared them to support others' needs on demand between their own goals. Spectators reported that the priming and cues in CrowdCheer prepared them to cheer for both their runner and other runners

once they had committed to an opportunity to do so. For example, when asked if they had enough information to spot and cheer for runners, spectators reported on average a 6.7 out of 7 (between agree and strongly agree, SD = 0.83). Spectators described how the static context (e.g. identifiable information) was useful for preparing them to cheer on-demand. In the post-survey, Spectator D wrote that seeing "their photo and that they were nearby" were details that helped her cheer for another runner. Spectators also described how dynamic context (e.g. the distance tracker and visual cues) was useful for cueing them to focus at different moments. In her interview, Spectator A said: *"I loved seeing the distance he was away from me change colors. I was like, okay he's getting close, he's getting really close. It was like a push, like "hey, you need to start focusing on your runner."* In her interview, Spectator B reflected on how she relied on both static and dynamic context to prepare to cheer for a runner she did not know, saying: *"[it was] pretty cool being able to spot them based on their description. When I saw her, I don't know... I think I just happened to glance down at the app and saw that she was close, and then looked up and then like a minute or so later and I just recognized her."* In her interview, Spectator C described how her preparedness to cheer for her own runner differed from her typical spectator experience, saying that she *"liked the accuracy of the app – we were in the right place for them to see us. Normally, you're guessing."* In contrast to our needfinding where gathering context for real-time support required advanced planning or extra monitoring effort, these findings suggest that continual support systems can prime and cue individuals in ways that prepare them to both provide support on-demand and focus on goals when needed.

6.3 Individuals enjoyed and valued supporting others' needs between their own goals

Generally, individuals found that they enjoyed having opportunities to support the needs of others in the moments they were not focusing on their own goals. Specifically, spectators reported that they generally enjoyed using CrowdCheer to cheer for other runners during the race (5.6 out of 7, between somewhat agree and strongly agree, SD: 1.14). In the post-survey, Spectator A wrote "It showed me people who were around and gave me the option to cheer for them as well. It was pretty cool." When asked as a follow up question in an interview about what it felt like to see runners from the app positively respond when they cheered, Spectator A said: *"It was exciting! It felt good! It was like they knew someone – even though they didn't know me – but I think it's like that they recognize their cheers, so it was good to know that even some strangers had their support, and it felt good to be out there and support."* Spectator D described how being able to see explicit opportunities to directly cheer for other runners helped her to stay engaged, in contrast to activities she would typically do such as check her phone: In her interview, she said *"Like, I don't want to miss [her runner's name], but seeing these other runners, feeling engaged. [...] It was really cool to have an opportunity to support both [her runner's name] and other people."* These findings suggest that by creating an opportunity for individuals to support others' needs while still being able to focus on their primary goals, continual support systems can activate these potential helpers who are generally willing, available, and able to help, but face challenges in being able to balance both outcomes.

Individuals in this ad-hoc crowd setting also described why they valued supporting individuals they did not know between the moments where they needed to focus on their own goals. When reflecting on their reasons that motivated them to support other runners, spectators in the ad-hoc crowd cited reasons such as "showing sportsmanship," "to encourage runners," and that "everyone can use support." In their interviews, multiple spectators also reflected on their previous experiences receiving support as runners, and used it to express the impact and meaning of their support at the race. For example, Spectator A said: *"I know how difficult it is as a runner to stay in it especially when it's so cold. For me it was just 'you got this' to see someone pump you up, it helps you [...] I thought about how it feels for other runners to cheer for me. They don't know me but they're cheering me on*

and it felt good. Those cheers make something hard, easier!" Spectator B also reflected on her past experiences, saying *"So I have been, in the past, a runner, so I know that even along the course, having people cheer for you even if you don't know who they are is very encouraging [...] and it's always very emotional for some reason, you know, to cheer for runners. [...] Depending on the type of runner you are, for some people a 5k is just an easy little run, but for others, it's something they train for and they work really hard at, you know. And just wanting to give them that little extra push to finish strong is kind of a cool thing."* These findings suggest that continual support systems can not only work in community settings where individuals may be bonded through social ties, but can be used to engage an ad-hoc crowd who may not personally know others, but are still willing and even desire to support people who face a challenge similar to one they may have supported previously, or have personally experienced in the past.

7 DISCUSSION

Our pilot study findings demonstrate how continual support systems can help individuals balance supporting the needs of others with focusing on their own goals with ease. Results from our pilot study show initial evidence for how a continual support system enables potential helpers to maintain awareness of, evaluate, and prepare for moments where goals demand attention or needs arise by (1) monitoring the goals of helpers and the needs of help-seekers, (2) matching potential helpers to opportune moments to support needs or focus on goals, and (3) priming helpers to prepare for providing on-demand support between goals. Finally, our results show that continual support systems are capable of activating these potential helpers within communities who value, desire, and enjoy supporting community needs between their own goals, but face challenges in doing so. These findings suggest how continual support systems can create sustainable opportunities for individuals to support the needs of others in ways that have the potential to scale the needs that can be met with existing community resources, and even enable increasingly supportive communities.

7.1 Continual Support Integrates CSCW and Crowd Computing Approaches

Continual support can meet dynamic and on-demand needs with existing community resources because it integrates how CSCW and crowd computing research have approached systems for supporting needs in the past. Prior CSCW research has explored systems that better connect people's needs to existing resources in their community [26, 28]; however, these systems lack the models of changing needs and goals needed to connect a need to an available helper on-demand. Prior crowd computing research has explored systems that enable real-time support via on-demand crowds [5] and convenient tasks for volunteer helpers [21]; however, these systems do not consider how a helper's capacity and willingness to contribute support in the moment can depend on the state of their own goals. Across CSCW and crowd computing, little work has explored how we might directly enable potential helpers to balance supporting emerging needs of others with focusing on their own goals within a community. The continual support approach is designed to address these gaps by integrating approaches that explicitly monitor people's needs and goals, and coordinate support in ways that are cognizant of and directly support the goals of potential helpers in the community. Future crowd support systems can take inspiration from helpseeking research to extend beyond convenience in task design and explicitly integrate and support individual's goals in their models. Similarly, future helpseeking systems can take inspiration from crowd computing research to more deeply integrate technical approaches to tracking goals and optimally coordinating real-time help across a community in scalable ways. By integrating these approaches, we can overcome the social-technical gap [1] in leveraging available community resources to continually meet emerging needs on-demand, and design frameworks like continual support that have the potential to sustainably

scale the needs that can be met with existing community resources, thus enabling increasingly supportive communities.

7.2 Coordinating Support as Needs and Goals Scale

In larger community settings or situations where individuals have many more needs and goals, we may need to extend continual support systems to coordinate support in scalable ways that best satisfy both community and individual helper outcomes. Our findings demonstrate that identifying opportune moments for ad-hoc crowds of spectators to support other runners was highly effective for meeting more community needs with existing help resources. However, as the number of needs and goals scale and communities grow, there may be scenarios where certain needs are more urgent than others, or times when the system may want to be strategic about distributing help across those available in the community. For example, as runners scale, there may be many needs for the ad-hoc crowd of spectators to support. Potential helpers may need support in filtering and focusing on particular opportunities based on factors like who is best equipped to support a particular runner, or which runners haven't been cheered in a while. As spectators scale, systems may consider approaches for distributing effort or improved needs coverage [12], for example by directing opportunities towards spectators who may have engaged in fewer opportunities than others. In some cases, it may be better to postpone less urgent needs if the system anticipates that other helpers may have the opportunity to support those needs at a later time [21]. As individual helpers have more goals, continual support systems may need more advanced representations of goals that consider degrees of priority, like if a spectator has multiple primary runners that they want to cheer on at the race. As individuals with needs and goals scale, future continual support systems can integrate different coordination algorithms that consider various ways best satisfy the many needs in a community given existing support resources. Further, such systems may even surface the tradeoffs of different approaches and potential consequences for individuals and the broader community, allowing people to evaluate what they may want for both as individuals and as the community at large.

7.3 Studying Continual Support in Diverse Settings

While we focused our pilot evaluation on an ad-hoc crowd of spectators, the challenges that the ad-hoc crowd face may persist across many different settings where individuals are generally willing, available, and able to support others' needs but struggle to balance this with their own goals when needs emerge and the status of their own goals change continuously. As an illustrative example, a software engineer in a workplace environment may be generally willing to help their colleagues debug issues, available to do so between their own deployments, and have the expertise required to offer that help. However, the software engineer may struggle to keep track of the status of their own development tasks at the same time as emerging debugging needs from their colleagues. The software engineer may also struggle to evaluate when to help a colleague with a particular debug need and when to focus on their own deliverables. Even if the software engineer identifies a window where they can help a particular need, the software engineer may have the general Javascript expertise needed to help, but may not be familiar with the particular codebase or libraries that their colleague is using. Without this project specific context, that software engineer may still be unprepared to offer the most effective help in the moment where their availability aligns with their colleague's need. As another illustrative example, students leading their own projects (e.g. design projects) in a studio-based learning environment may be generally willing, available, and able to support the needs of their peers. However, a student may still struggle to keep track of their own project progress and when their peers need help. The student may also struggle to evaluate whether they have the capacity to offer help in those moments based on the

status of their own project. Further, the student may have relevant prototyping expertise, but may still lack the context about the other person's project, leaving them unprepared to offer effective help in the moments where they do have availability. Moving forward, we may want to explore the application of the continual support framework in other settings like workplace and learning environments as a way to create feasible support opportunities for individuals to meet more needs within their communities.

7.4 Handling Diverse and Complex Needs and Goals with Mixed-Initiative Approaches

Further, continual support systems may need to incorporate mixed-initiative approaches [17] to handle diverse and complex sets of needs and goals that often occur in other community settings, but are challenging to model and monitor. Our findings demonstrate that modeling and monitoring the emerging support opportunities and primary goals for spectators is highly effective for enabling an ad-hoc crowd to provide support. However, as community needs and individuals' goals grow complex and diversify (e.g. in workplace and learning settings), the types of needs and goals could vary significantly from person to person, making it hard to model and track them automatically. For example, there may be complex needs in learning and workplace settings that are harder to track (such as when a student may be struggling) and require tacit knowledge of community members to better diagnose (such as the expertise of a mentor). Similarly, a person may have other goals that are not initially monitored or modeled automatically, but still affect their ability to provide help, such as the case where a spectator was also watching her nephew. While our model did not include represent all goals, users still had flexibility in evaluating and making decisions for themselves about when to support, rather than being assigned support tasks. In such settings where we may not be able to automatically and exhaustively monitor and model needs and goals, it's important that future continual support systems can extend to enable mixed-initiative approaches that supplement automated models with human input when automated models may fail.

REFERENCES

[1] Mark S Ackerman. 2000. The intellectual challenge of CSCW: the gap between social requirements and technical feasibility. *Human–Computer Interaction* 15, 2-3 (2000), 179–203.

[2] Mark S. Ackerman, Juri Dachtera, Volkmar Pipek, and Volker Wulf. 2013. Sharing Knowledge and Expertise: The CSCW View of Knowledge Management. *Computer Supported Cooperative Work (CSCW)* 22, 4 (01 Aug 2013), 531–573. https://doi.org/10.1007/s10606-013-9192-8

[3] BOA Chicago Marathon App. 2019. (2019). https://www.chicagomarathon.com/runners/runner-information/#app

[4] TCS New York City Marathon App and Runner Tracking. 2019. (2019). https://www.tcsnycmarathon.org/race-day/app-and-runner-tracking

[5] Michael S. Bernstein, Joel Brandt, Robert C. Miller, and David R. Karger. 2011. Crowds in Two Seconds: Enabling Realtime Crowd-powered Interfaces. (2011), 33–42. https://doi.org/10.1145/2047196.2047201

[6] Jeffrey P. Bigham, Chandrika Jayant, Hanjie Ji, Greg Little, Andrew Miller, Robert C. Miller, Aubrey Tatarowicz, Brandyn White, Samuel White, and Tom Yeh. 2010. VizWiz: Nearly Real-time Answers to Visual Questions. , Article 24 (2010), 2 pages. https://doi.org/10.1145/1805986.1806020

[7] Mark Bilandzic, Marcus Foth, and Alexander De Luca. 2008. CityFlocks: designing social navigation for urban mobile information systems. In *Proceedings of the 7th ACM conference on Designing interactive systems*. ACM, 174–183.

[8] Carrie J Cai, Shamsi T Iqbal, and Jaime Teevan. 2016. Chain reactions: The impact of order on microtask chains. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 3143–3154.

[9] Franco Curmi, Maria Angela Ferrario, and Jon Whittle. 2017. Embedding a Crowd Inside a Relay Baton: A Case Study in a Non-Competitive Sporting Activity. (2017), 2359–2370. https://doi.org/10.1145/3025453.3025551

[10] Franco Curmi, Maria Angela Ferrario, Jon Whittle, and Florian 'Floyd' Mueller. 2015. Crowdsourcing Synchronous Spectator Support: (Go on, Go on, You'Re the Best)N-1. (2015), 757–766. https://doi.org/10.1145/2702123.2702338

[11] Matthew W. Easterday, Daniel G. Rees Lewis, and Elizabeth M. Gerber. 2017. The logic of design research. *Learning: Research and Practice* (2017), 1–30. https://doi.org/10.1080/23735082.2017.1286367

[12] Kapil Garg, Yongsung Kim, Darren Gergle, and Haoqi Zhang. 2019. 4X: A Hybrid Approach for Scaffolding Data Collection and Interest in Low-Effort Participatory Sensing. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article Article 90 (Nov. 2019), 28 pages. https://doi.org/10.1145/3359192

[13] Zhiwei Guan, Shirley Lee, Elisabeth Cuddihy, and Judith Ramey. 2006. The Validity of the Stimulated Retrospective Think-aloud Method As Measured by Eye Tracking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*. ACM, New York, NY, USA, 1253–1262. https://doi.org/10.1145/1124772.1124961

[14] Emily Harburg, Yongsung Kim, Elizabeth Gerber, and Haoqi Zhang. 2015. CrowdFound: A Mobile Crowdsourcing System to Find Lost Items On-the-Go. (2015), 1537–1542. https://doi.org/10.1145/2702613.2732757

[15] Damon Horowitz and Sepandar D Kamvar. 2010. The anatomy of a large-scale social search engine. In *Proceedings of the 19th international conference on World wide web*. ACM, 431–440.

[16] Eric Horvitz. 1997. Models of Continual Computation. (1997), 286–293. http://dl.acm.org/citation.cfm?id=1867406.1867451

[17] Eric Horvitz. 1999. Principles of Mixed-Initiative User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*. Association for Computing Machinery, New York, NY, USA, 159–166. https://doi.org/10.1145/302979.303030

[18] Eric Horvitz. 2001. Principles and applications of continual computation. *Artificial Intelligence* 126, 1 (2001), 159 – 196. https://doi.org/10.1016/S0004-3702(00)00082-5 Tradeoffs under Bounded Resources.

[19] BAA Boston Marathon Spectator Information. 2019. (2019). https://www.baa.org/races/boston-marathon/enter/att-athlete-alerts

[20] Yongsung Kim. 2015. Libero: On-the-go Crowdsourcing for Package Delivery. (2015), 121–126. https://doi.org/10.1145/2702613.2726964

[21] Yongsung Kim, Darren Gergle, and Haoqi Zhang. 2018. Hit-or-Wait: Coordinating Opportunistic Low-effort Contributions to Achieve Global Outcomes in On-the-go Crowdsourcing. , Article 96 (2018), 12 pages. https://doi.org/10.1145/3173574.3173670

[22] Kristina Knaving, PawełWoźniak, Morten Fjeld, and Staffan Björk. 2015. Flow is Not Enough: Understanding the Needs of Advanced Amateur Runners to Design Motivation Technology. (2015), 2013–2022. https://doi.org/10.1145/2702123.2702542

[23] Walter Lasecki, Christopher Miller, Adam Sadilek, Andrew Abumoussa, Donato Borrello, Raja Kushalnagar, and Jeffrey Bigham. 2012. Real-time Captioning by Groups of Non-experts. (2012), 23–34. https://doi.org/10.1145/2380116.2380122

[24] 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. (2014), 551–562. https://doi.org/10.1145/2642918.2647367

[25] David W. McDonald. 2001. Evaluating Expertise Recommendations. In *Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work (GROUP '01)*. Association for Computing Machinery, New York, NY, USA, 214–223. https://doi.org/10.1145/500286.500319

[26] David W. McDonald and Mark S. Ackerman. 2000. Expertise Recommender: A Flexible Recommendation System and Architecture. (2000), 231–240. https://doi.org/10.1145/358916.358994

[27] Robert Miller. 2016. The Problem of Context in Microtasks. (2016).

[28] Robert C. Miller, Haoqi Zhang, Eric Gilbert, and Elizabeth Gerber. 2014. Pair Research: Matching People for Collaboration, Learning, and Productivity. (2014), 1043–1048. https://doi.org/10.1145/2531602.2531703

[29] Jeffrey Nichols and Jeon-Hyung Kang. 2012. Asking questions of targeted strangers on social networks. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*. ACM, 999–1002.

[30] Akshay Rao, Harmanpreet Kaur, and Walter S. Lasecki. 2018. Plexiglass: Multiplexing Passive and Active Tasks for More Efficient Crowdsourcing. (2018).

[31] Andrzej Romanowski, Sven Mayer, Lars Lischke, Krzysztof Grudzień, Tomasz Jaworski, Izabela Perenc, PrzemysKucharski, Mohammad Obaid, Tomasz Kosizski, and Paweł W. Wozniak. 2017. Towards Supporting Remote Cheering During Running Races with Drone Technology. (2017), 2867–2874. https://doi.org/10.1145/3027063.3053218

[32] Slack. 2019. (2019). https://slack.com/

[33] Jan Van den Akker. 1999. Principles and methods of development research. In *Design approaches and tools in education and training*. Springer, 1–14.

[34] Ryen W White, Matthew Richardson, and Yandong Liu. 2011. Effects of community size and contact rate in synchronous social q&a. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2837–2846.