Shepherding the Crowd: An Approach to More Creative Crowd Work

Steven P Dow & Scott R Klemmer
Stanford HCI Group
[spdow, srk]@stanford.edu

ABSTRACT

Micro-task platforms provide a marketplace for hiring people to do short-term work for small payments. Requesters often struggle to obtain high-quality results, especially on content-creation tasks, because work cannot be easily verified and workers can move to other tasks without consequence. Such platforms provide little opportunity for workers to reflect and improve their task performance. Timely and task-specific feedback can help crowd workers learn, persist, and produce better results. We analyze the design space for crowd feedback and introduce *Shepherd*, a prototype system for visualizing crowd work, providing feedback, and promoting workers into shepherding roles. This paper describes our current progress and our plans for system development and evaluation.

Author Keywords
Innovation, creativity, feedback, critique, managing crowds

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

How can we educate and motivate online distributed workforces to accomplish more creative and complex projects? To understand the mechanics of large-scale creative work, our research examines how individual and small team design practices affect results. Our experiments empirically demonstrate that simple process changes can help people design better solutions. For example, creating and receiving feedback on multiple design ideas in parallel, as opposed to serially, leads people to produce more diverse, better solutions [6]. Furthermore, parallel prototypers react more positively to critique and share more fluidly with group members [5]. A key methodological insight in this research has been challenging participants to do tasks where the solutions are both creatively different and objectively measurable – like creating Web advertisements. For these experiments, crowdsourcing has proven to be invaluable for obtaining judgments of design quality and divergence. As a measure of design diversity, Mechanical Turk raters assessed the pair-wise similarity of all combinations of participant ads. Human judgments and Web analytics offer powerful measures for examining the active ingredients behind human creativity and teamwork.

Building on this theoretical understanding of the cognitive and social mechanics of design practice, we are currently exploring how to support more innovative work in distributed micro-task platforms. We propose two key features will help modern micro-task platforms accomplish more complex and creative work. First, formal feedback will help workers learn tasks and keep them motivated. Second, real-time visualizations of completed tasks will provide requesters a means to monitor and shepherd workers. For workers, an holistic view of tasks may motivate workers to contribute in more ways to the project. We hypothesize that providing infrastructural support for formal critique and worker interaction will lead to better educated, more motivated workers, and better work results.

MOTIVATION AND BACKGROUND

On micro-task platforms such as Mechanical Turk (www.mturk.com), requesters pay people to execute short tasks for small amounts of money. Unlike peer-production systems, requesters and workers remain largely anonymous to each other, and little direct interaction occurs between them. Workers can only communicate with other workers through third-party forums (http://turkopticon.differenceengines.com). From a labor perspective, treating people as interchangeable replacements for computational processes means that workers often submit assignments with minimal effort [9], and have little opportunity or motivation to improve their understanding of a task domain.

For simple tasks such as data entry, requesters can validate work quality by redundantly hiring workers for the same job [8] or by inserting test problems that have known solutions [9]. However, these strategies are less effective for content-creation tasks — such as writing product reviews, designing advertisements, or categorizing complex data — where requesters desire original and diverse content.

One strategy for accomplishing more complex work is to decompose tasks into iterative or parallel subtasks [3,12]. Soylent introduced a find-fix-verify pattern for word processing, where different workers each take on a smaller piece of the larger task [3]. However, within those smaller tasks, an underlying problem persists:

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2011, May 7–12, 2011, Vancouver, BC, Canada.
Copyright 2011 ACM 978-1-4503-0267-8/11/05....$10.00.

workers are not encouraged to learn or improve their performance. How can crowdsourcing platforms motivate and scaffold novice workers to improve over time, especially on complex, large-scale, creative tasks? We hypothesize that worker interaction with requesters and with other workers is a key missing component.

In many communities of practice, senior members (often implicitly) help novices learn and stay motivated [11]. Traditional work environments foster employee development through formal performance reviews and feedback, and informally, through peripheral participation [11]. Online communities often provide infrastructure for moderators to review others' content and to encourage the growth of newer members [10]. Peer-production projects like Wikipedia and open-source software have decentralized rather than hierarchical management systems [2]. Individuals choose where to devote resources, and through transparency and reputation systems, the community defines standards and quality control mechanisms [14].

In contrast with traditional firms or peer-production systems, micro-task platforms such as Amazon Mechanical Turk typically offer few formal or informal methods for worker-requester communication. Instructions provide the primary point of contact. The products of crowd workers are invisible to peers. As a result, novice workers cannot observe expert behavior. From a learning perspective, social interaction provides an essential form of feedback [1]. Peer interaction also has motivational benefits [4,7]. LiveOps, a distributed online call center, enabled chat interaction between at-home agents to recreate a "water cooler" setting and to foster cohesion among their workforce [13].

Interactive feedback complements other quality-improvement efforts such as worker qualifications and clearer instructions. We hypothesize that task-specific feedback will help workers on micro-task markets improve performance, much as it does in real-world settings, and make workers cognizant that their work is under review. Additionally, feedback may motivate workers to persevere and accept additional tasks. We investigate these hypotheses through a prototype system, Shepherd, that demonstrates how to make feedback an integral part of crowdsourced creative work.

OPPORTUNITIES FOR CROWD FEEDBACK

To effectively design feedback mechanisms that achieve the goals of learning, engagement, and quality improvement, we first analyze the important dimensions of the design space for crowd feedback (Figure 1).

Timeliness: When should feedback be shown?

In micro-task work, workers stay with tasks for a short while, then move on. This implies two timing options: synchronously deliver feedback when workers are still engaged in a set of tasks, or asynchronously deliver feedback after workers have completed the tasks.

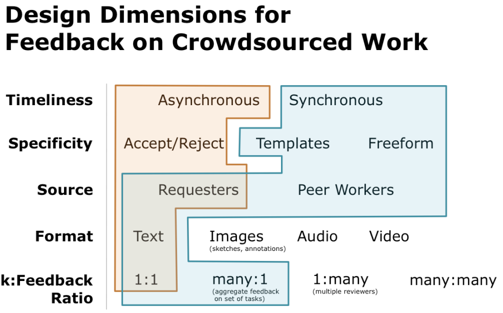


Figure 1: Current systems (in orange) focus on asynchronous, single-bit feedback by requesters. *Shepherd* (in blue) investigates richer, synchronous feedback by requesters and peers.

Synchronous feedback may have more impact on future task performance since it arrives while workers are still thinking about the task domain. It also increases the probability that workers will continue onto similar tasks. However, synchronous feedback places a burden on the feedback providers; they have little time to review work. This implies a need for tools or scheduling algorithms that enable near real-time feedback. Asynchronous feedback gives feedback providers more time to review and comment on work. However, workers may have forgotten about the task or feel unmotivated to review the feedback and to return to the task.

Currently, platforms like Mechanical Turk only allow asynchronous feedback with no enticement to return. Requesters can provide feedback at payment time, but at that point (typically days later), workers care more about getting paid than improving submitted work. More importantly, unless requesters have more jobs available, workers cannot act on requesters' advice.

Specificity: How detailed should feedback be?

Mechanical Turk currently allows requesters one bit of feedback—accept or reject. While additional freeform communication is possible, it is rarely used unless workers file complaints. Workers may learn more if they receive detailed and personalized feedback on each piece of work. However, this added specificity comes at a price: feedback providers must spend time authoring feedback. When feedback resources are limited, customizable templates can accelerate feedback generation and enable requesters to codify domain knowledge into pre-authored statements. However, templates could be perceived as overly general or repetitive, reducing their desired impact. Workers may need explicit incentive to read and reflect on feedback.

Source: Who should provide feedback?

Crowdsourcing requesters post tasks with specific quality objectives in mind; they are a natural choice for assuming the feedback role. However, experts often underestimate the difficulty novices face in solving tasks [7] or use language or concepts that are beyond the grasp of novices [6]. Moreover, as feedback becomes more specific, requesters may find it more difficult to complete work assessments in real-time.

Alternatively, workers can be paid to provide feedback to other workers. Peer feedback increases scalability as more crowd workers can be recruited to handle the volume of feedback needs. Our preliminary trials indicate that workers perform tasks simultaneously and overlap (see Figure 2). In principle, this overlap opens up the possibility of peer feedback. For example, workers can be promoted into a feedback role after they successfully finish a series of tasks. This introduces the challenge of identifying and promoting knowledgeable and responsible workers.

SHEPHERD: SYSTEM DESIGN

We are developing Shepherd, an infrastructure for managing and providing feedback to crowd workers. Our vision is to make targeted feedback a core component of future micro-task platforms. Requesters will need interfaces to simultaneously author the task and associated feedback form. To administer feedback, requesters will need tools for visualizing work progress. The system will need to elegantly present feedback to workers and confirm that they see and understand the feedback. Also, the system should help requesters decide which workers to promote into advanced roles.

Current Progress

Our prototype recruits and pays workers through Amazon Mechanical Turk; task hosting and data collection occurs on our own Web server. Shepherd displays an overview of workers and results in real-time. The timeline view (Figure 2) presents a Gantt chart showing when workers accept a task, the length of time workers spend on each task, and how many tasks a worker completes within a batch. In the matrix view (Figure 3), columns show tasks and rows show workers. Each box shows the current state of a task (skipped, in progress, finished & needs feedback, or feedback applied).

Requesters can monitor incoming work and click on any task to provide feedback using specially designed forms. To streamline the process, the requester checks high-level feedback categories and the worker receives corresponding critique statements. By default, the system delivers feedback just before a worker begins a new task from the same batch. The choice about timing and delivery method is an empirical question, and depends on factors such as task type and scale.

Future Development

Micro-task platforms typically provide task authoring templates. Shepherd will give requesters tools for specifying feedback forms in tandem with task creation. Feedback templates become especially important when workers review others' work. We will evaluate the overhead costs for creating feedback templates in addition to the task.

A workforce administration interface will let requesters promote/demote workers to shepherding roles, track worker performance over time, and launch tasks for specific workers under controlled criteria. An inference algorithm will recommend promising workers based on prior task per-

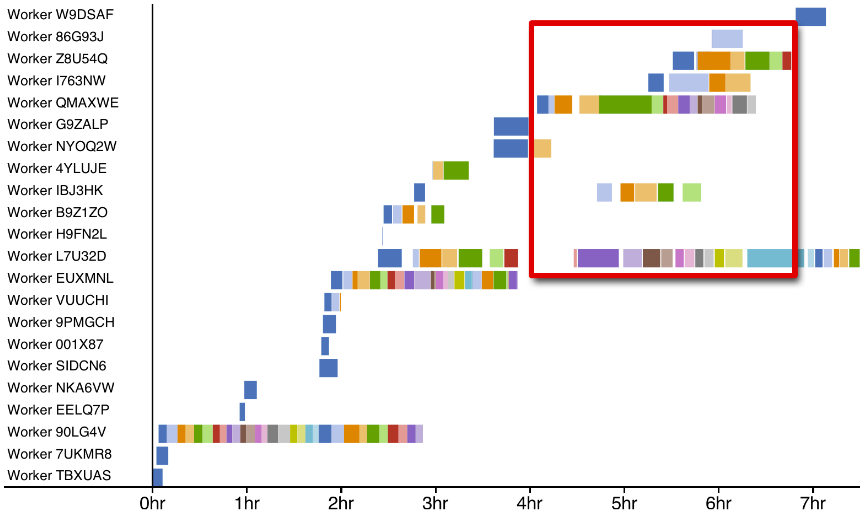
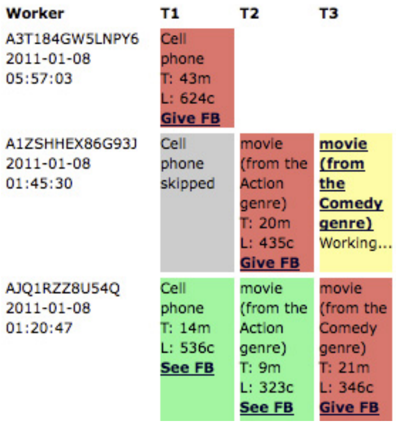


Figure 2: *Shepherd*'s timeline view. Workers overlap in time, which shows potential for peer feedback. This visualization shows work times for 100 product reviews. Rows represent individual workers. The X axis shows time. Each colored bar is one product review. The red rectangle highlights a time segment with significant overlap: multiple workers are active simultaneously.



Worker	T1	T2	T3
A3T184GW5LNPY6 2011-01-08 05:57:03	🟥 Cell phone T: 43m L: 624c Give FB		
A1ZSHHEX86G93J 2011-01-08 01:45:30	▨ Cell phone skipped	🟥 movie (from the Action genre) T: 20m L: 435c Give FB	🟨 movie (from the Comedy genre) Working...
AJQ1RZZ8U54Q 2011-01-08 01:20:47	🟩 Cell phone T: 14m L: 536c See FB	🟩 movie (from the Action genre) T: 9m L: 323c See FB	🟥 movie (from the Comedy genre) T: 21m L: 346c Give FB

Figure 3: *Shepherd*'s matrix view for a batch of product review tasks. Each box represents the current state of a task. Tasks can be completed in parallel by multiple workers (rows). Red boxes indicate tasks are ready for review. Yellow boxes are tasks in progress. Green boxes indicate that work is finished and feedback provided. Grey boxes show tasks that workers choose to skip.

formance and domain knowledge ascertained from short interspersed test questions.

POTENTIAL FOR CREATIVE CROWD WORK

Does the added cost of assessing work outweigh simpler mechanisms such as asking workers to assess their own work? We are currently working on an experiment to compare requester-provided and self-report assessments. Participants will write customer reviews for products or services. In this common crowdsourcing task, workers can potentially benefit from expert feedback. We can measure performance by hosting reviews on product sites and measuring community feedback on their helpfulness. Our study will also analyze overhead costs associated with providing feedback; worker self-assessments may lead to cheaper performance gains.

Longer term, we want to investigate the potential of recruiting workers to provide feedback for other workers on a large-scale content-creation project. We will study differences in how workers and requesters confer feedback and examine the effects of the presentation, source, and tone of feedback.

What's the broader potential for crowd creativity? Online crowds could help satisfy demand for personalized versions of artifacts. Example tasks may include customizing a greeting card for a particular demographic, generating bumper sticker ideas for local events, creating a mobile phone case design for Justin Bieber fans, etc. Crowds with feedback could effectively handle a large quantity of small personalization tasks. For more complex projects, workers can contribute to multiple different tasks and transition to roles with more responsibility. Further, a holistic visualization of a project and its various subtasks may motivate crowd workers. Workers would ideally be able to see how they contributed to the whole; this visibility is one key to success for projects like Wikipedia and The Johnny Cash Project.

AUTHOR BIOS

Steven P Dow is a Postdoctoral Scholar in the HCI Group at Stanford University where he researches human-computer interaction, creative problem-solving, prototyping practices, and crowdsourcing. Steven taught Stanford's first-ever research seminar on crowdsourcing. He is recipient of Stanford's Postdoctoral Research Award and co-recipient of a Hasso Plattner Design Thinking Research Grant. He received an MS and PhD in Human-Centered Computing from the Georgia Institute of Technology, and a BS in Industrial Engineering from University of Iowa. http://www.stanford.edu/~spdow/

Scott R Klemmer is an Assistant Professor of Computer Science at Stanford University, where he co-directs the Human-Computer Interaction Group. Organizations around the world use his lab's open-source design tools, and several books and popular press articles have covered his research. He is a co-recipient of a best paper award at both of the premier human-computer interaction conferences (CHI and UIST), Microsoft Research New Faculty Fellowship, Sloan Fellowship, and NSF CAREER award. He received a dual BA in Art-Semiotics and Computer Science from Brown University, and an MS and PhD in Computer Science from UC Berkeley. http://hci.stanford.edu/srk/

ACKNOWLEDGEMENTS

We want to thank our colleagues Björn Hartmann, Anand Kulkari, Truc Nguyen, and Brie Bunge for their hard work and contributions to this project.

REFERENCES

1. Annett, J. Feedback and human behaviour: the effects of knowledge of results, incentives, and reinforcement on learning and performance. Penguin Books, 1969.
2. Benkler, Y. Coase's Penguin, or, Linux and "The Nature of the Firm."The Yale Law Journal 112, 3 (2002), 369-446.
3. Bernstein, M.S., Little, G., Miller, R.C., et al. Soylent: a word processor with a crowd inside. Proceedings of the 23nd annual ACM symposium on User interface software and technology, ACM (2010), 313–322.
4. Cheshire, C. and Antin, J. The Social Psychological Effects of Feedback on the Production of Internet Information Pools. Journal of Computer-Mediated Communication 13, 3 (2008), 705-727.
5. Dow, S.P., Fortuna, J., Schwartz, D., Altringer, B., Schwartz, D.L., and Klemmer, S.R. Prototyping Dynamics: Sharing Multiple Designs Improves Exploration, Group Rapport, and Results. Conf on Human Factors in Computing Systems, (2011).
6. Dow, S., Glassco, A., Kass, J., Schwarz, M., Schwartz, D.L., and Klemmer, S.R. Parallel Prototyping Leads to Better Design Results, More Divergence, and Increased Self-Efficacy. Transactions on Computer-Human Interaction 4, (2010).
7. Horton, J.J. Employer Expectations, Peer Effects and Productivity: Evidence from a Series of Field Experiments. SSRN eLibrary, (2010).
8. Ipeirotis, P.G., Provost, F., and Wang, J. Quality management on Amazon Mechanical Turk. Proceedings of the ACM SIGKDD Workshop on Human Computation, ACM (2010), 64–67.
9. Kittur, A., Chi, E.H., and Suh, B. Crowdsourcing user studies with Mechanical Turk. Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, ACM (2008), 453-456.
10. Lampe, C. and Resnick, P. Slash(dot) and burn: distributed moderation in a large online conversation space. Proceedings of the SIGCHI conference on Human factors in computing systems, ACM (2004), 543–550.
11. Lave, J. and Wenger, E. Situated Learning: Legitimate Peripheral Participation. Cambridge University Press, 1991.
12. Little, G., Chilton, L.B., Goldman, M., and Miller, R.C. TurKit: tools for iterative tasks on mechanical Turk. Proceedings of the ACM SIGKDD Workshop on Human Computation, ACM (2009), 29–30.
13. Musico, C. There's No Place Like Home. destinationCRM.com, 2008.
14. Viégas, F., Wattenberg, M., and Mckeon, M. The Hidden Order of Wikipedia. In Online Communities and Social Computing. 2007, 445-454.