

Enabling Expert Crowdsourcing with Flash Teams

Daniela Retelny, Sébastien Robaszkiewicz, Alexandra To, Michael Bernstein

Stanford University HCI Group

{dretelny, robi, ato1120, msb}@cs.stanford.edu

This abstract summarizes work that is in preparation for submission at CHI 2014.

Abstract

Crowdsourcing enables individuals to come together quickly and complete projects that would be virtually impossible for a single individual to accomplish at the same scale [1,2]. To date, the dominant paradigm has been to combine many paid non-expert opinions, for example from Amazon Mechanical Turk,¹ to match the performance of a single expert [2,4]. However, this paradigm is plateauing in complexity: many creative, open-ended or highly complex tasks remain largely unsolved because they require significant expertise that cannot be designed into the system [3,5]. Our objective is to drive crowdsourcing research past this complexity boundary by *introducing expert teams as core elements of crowdsourcing systems and developing modular computational techniques to guide these teams*.

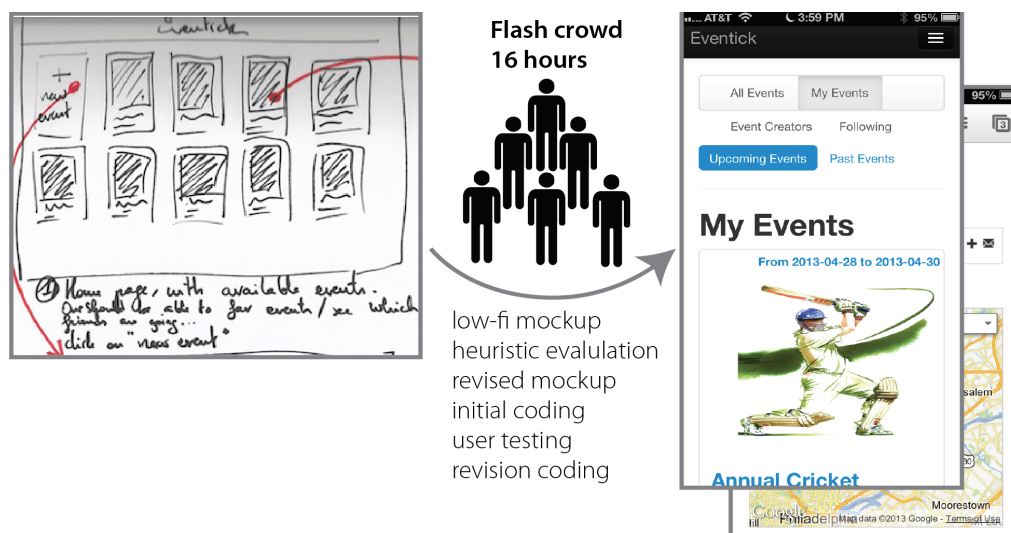


Figure 1. Flash crowdsourcing guides teams of distributed, crowdsourced experts to take an initial napkin sketch, run it through a user-centered design process, and produce a user-tested, working application in one day.

This research addresses the complexity challenge in expert crowdsourcing by introducing *flash teams*: modular computational workflows that coordinate experts from the crowd and enable them to complete complex and interdependent goals (Figure 1). Modular design means that each flash team is composable, reusable and self-contained, with a defined structure for how the crowd experts interact (Figure 2). Modularity also allows for flash teams to be replicated and combined to create ad-hoc flash organizations (Figure 2, right). To address effectiveness, flash teams capitalize on the unique strengths of crowdsourcing relative to traditional distributed teams: we introduce *elastic* recruiting techniques that allow teams to grow and shrink on demand, and *pipelining* to accelerate completion times by beginning the next stage of a workflow as soon as the previous stage has sufficient preliminary results.

The motivating question for flash teams: what would it take to begin with a design sketch on a

¹ <http://www.mturk.com>

napkin and gather an expert crowd to follow the entire user-centered design process — a designer to mock up interface alternatives, a usability analyst to test those prototypes, and a front-end engineer to implement the best one — all in a single day (Figure 1)? Or to create an animated short movie in 20 hours? Or to develop an entire online learning (MOOC) platform and a series of lessons in 36 hours? We hypothesize that expert crowds could accomplish these goals if we provide an approach to manage the complexity of distributed crowd work.

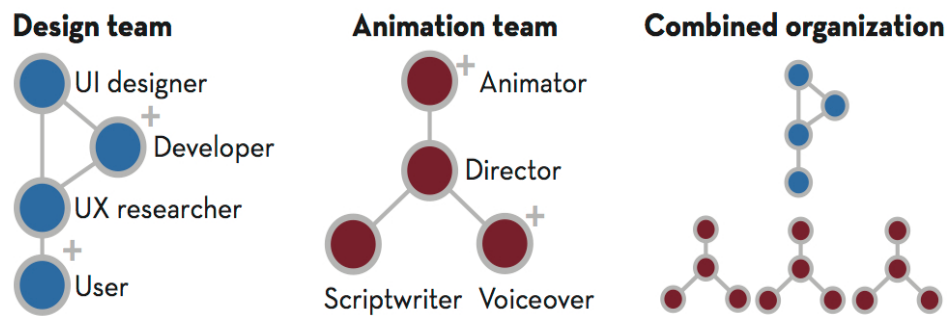


Figure 2. Modular team abstractions for flash expert teams. For example, a design team has one or more (+) developers, a UI designer, a UX researcher, and one or more (+) users, each with coordination dependencies indicated via edges. These teams are composable: a large team creating on-demand MOOCs might combine a design team for the technical content with multiple animation teams, one for each type of lesson.

Flash teams suggest that user interfaces and computational crowdsourcing techniques can play a central role in expert crowdsourcing. In this research, we demonstrate that flash teams can solve a wide range of complex tasks by presenting flash team structures that illustrate composability, replicability and elasticity on tasks such as designing user-tested software from a napkin sketch, creating educational content for a course, and developing a short animation. We also present *Foundry*, an authoring platform that combines the visual language of team workflow environments with the affordances of flash teams in order to aid users in composing modular, elastic and pipelined team designs. Our vision is for Foundry to become a library of best practices, as well as a first-generation IDE for expert crowd computing.

References

1. Bernstein, M.S., Little, G., Miller, R.C., et al. Soylent: A Word Processor with a Crowd Inside. *Proc. UIST '10*, (2010).
2. Kittur, A., Smus, B., Khamkar, S., and Kraut, R.E. CrowdForge. *Proc. UIST '11*, (2011).
3. Kittur, A., Nickerson, J., Bernstein, M., Gerber, E., Shaw, A., Zimmerman, J., Lease, M., and Horton, J. The Future of Crowd Work. *Proc. CSCW '13*, (2013).
4. Little, G., Chilton, L., Goldman, M., and Miller, R.C. Exploring iterative and parallel human computation processes. *Proc. HCOMP '10*, (2010)
5. Yu, L. and Nickerson, J. V. Cooks or cobblers?: crowd creativity through combination. *Proc. CHI '11*, (2011).