The Problem of Context in Microtasks

Rob Miller
MIT CSAIL
Cambridge, MA 02139, USA
rcm@mit.edu

Context and Task State in Microtasks

One key challenge in decomposing a problem into microtasks is providing sufficient context for each microtask. Some of this context is knowledge in the worker's head, which must be written down for the benefit of other workers. This is sometimes an advantage: micro-outsourcing [12] has noted that, at least in software development, the need to write the context down for microtask workers could make the code more maintainable in the long run. Sometimes it's even good to lose this mental context; selfsourcing [14] and the crowd within [15] suggest that doing a microtask repeatedly with different contexts can give better results.

Another kind of context is the set of tools needed to do the job, and the arrangement of the workspace. The usual crowdsourcing approach to tool context is a custom-built user interface for each kind of microtask, which is costly for one-off workflows. Another approach is an integrated environment that contains all the possible tools, regardless of the particular microtask to be done; examples include Mobi [11] for travel planning, Collabode [4] for software development, and Legion [13] for general-purpose UI manipulation.

A third kind of context is the overall state of the work – tasks that have already been done and work that remains to be done, in the form of plans, todo items, and outlines.

In general, we can't overlook the cost of saving and restoring context: writing and reading instructions, recalling the intent of a selfsourced task; collecting the tools you need and navigating to the right places in websites, documents, or code. We need to explore task decompositions and design patterns that minimize the cost of context.

Author Biography and Relevant Work

I am a professor of computer science at MIT, and a member of the Computer Science and Artificial Intelligence Laboratory (CSAIL). My research interests lie at the intersection of programming and human computer interaction, including crowd computing, online education, software development tools, and end-user programming. My research group has worked on crowdsourcing for eight years, primarily focusing on ways to integrate crowdsourcing into user interfaces. We built an early toolkit for crowd programming (Turkit [1]), integrated crowdsourced features into a desktop application (Soylent [2]), embedded a crowd into a realtime photo-taking interaction (Adrenaline [3]), and designed a collaborative IDE that supports *micro-outsourcing* of programming tasks (Collabode [4]). We have also studied the properties of iterative workflows [5], realtime crowds [6], and how workers find microtasks to do [7]. Our recent work has focused on massive online education, introducing *learnersourcing* [8,9] as a way to improve online educational materials, and *wait-learning* [10] as a way to fit microlearning into small dead moments of your day.

References

1. Greg Little, Lydia B. Chilton, Max Goldman, and Robert C. Miller. "TurKit: Human Computation Algorithms on Mechanical Turk." UIST 2010.
2. Michael S. Bernstein, Greg Little, Robert C. Miller, Bjoern Hartmann, Mark S. Ackerman, David R. Karger, David Crowell, and Katrina Panovich. "Soylent: A Word Processor with a Crowd Inside." UIST 2010.
3. Michael S. Bernstein, Joel Brandt, Robert C. Miller and David R. Karger. "Crowds in Two Seconds: Enabling Realtime Crowd-Powered Interfaces." UIST 2011.
4. Max Goldman, Greg Little, and Robert C. Miller. "Real-Time Collaborative Coding in a Web IDE." UIST 2011.
5. Greg Little, Lydia B. Chilton, Max Goldman, and Robert C. Miller. "Exploring iterative and parallel human computation processes." HCOMP 2010.
6. Michael Bernstein, David Karger, Rob Miller, Joel Brandt. "Analytic Methods for Optimizing Realtime Crowdsourcing." Collective Intelligence 2012.
7. Lydia B. Chilton, John J. Horton, Robert C. Miller, and Shiri Azenkot. "Task search in a human computation market." HCOMP 2010.
8. Juho Kim, Phu Nguyen, Sarah Weir, Philip J. Guo, Robert C. Miller, and Krzysztof Z. Gajos. "Crowdsourcing Step-by-Step Information Extraction to Enhance Existing How-to Videos." CHI 2014.
9. Sarah Weir, Juho Kim, Krzysztof Z. Gajos, and Robert C. Miller. "Learnersourcing Subgoal Labels for How-to Videos." CSCW 2015.
10. Carrie J. Cai, Philip J. Guo, James R. Glass, Robert C. Miller. "Wait-Learning: Leveraging Wait Time for Second Language Education." CHI 2015.
11. Haoqi Zhang, Edith Law, Robert C. Miller, Krzysztof Gajos, David Parkes, and Eric Horvitz. "Human Computation Tasks with Global Constraints." CHI 2012.
12. Max Goldman, Greg Little, Robert C. Miller. "Collabode: collaborative coding in the browser." CHASE 2011.
13. Walter S. Lasecki, Kyle I. Murray, Samuel White, Robert C. Miller, Jeffrey P. Bigham. "Real-time crowd control of existing interfaces." UIST 2011.
14. Jaime Teevan, Daniel Liebling, Walter Lasecki. "Selfsourcing personal tasks." CHI 2014 WIP.
15. E. Vul and H. Pashler. "Measuring the Crowd Within: Probabilistic Representations within Individuals." Pyschological Science, 19(7), 2008.