Selfsourcing Personal Tasks

Jaime Teevan
Microsoft Research
Redmond, WA 98052 USA
teevan@microsoft.com

Daniel J. Liebling
Microsoft Research
Redmond, WA 98052 USA
danl@microsoft.com

Walter S. Lasecki
University of Rochester
Rochester, NY 14627 USA
wlasecki@cs.rochester.edu

CHI 2014, April 26–May 1, 2014, Toronto, Ontario, Canada.
ACM 978-1-4503-2474-8/14/04.
http://dx.doi.org/10.1145/2559206.2581181

Abstract
Large tasks can be overwhelming. For example, many people have thousands of digital photographs that languish in unorganized archives because it is difficult and time consuming to gather them into meaningful collections. Such tasks are hard to start because they seem to require long uninterrupted periods of effort to make meaningful progress. We propose the idea of *selfsourcing* as a way to help people to perform large personal information tasks by breaking them into manageable microtasks. Using ideas from crowdsourcing and task management, selfsourcing can help people take advantage of existing gaps in time and recover quickly from interruptions. We present several achievable selfsourcing scenarios and explore how they can facilitate information work in interruption-driven environments.

Author Keywords
Selfsourcing, crowdsourcing, microtasks, interruption.

ACM Classification Keywords
H.5.m. Information interfaces and presentation: Misc.

Introduction and Related Work
The proverb, “A journey of a thousand miles begins with a single step,” is attributed to the sixth century BCE philosopher Laozi. People have attempted to accomplish large tasks by decomposing them into manageable parts for millennia, and modern approaches to time management continue to take Laozi’s words to heart. For example, agile software development breaks

Figure 1. Examples of microtasks for photo organization implemented on a mobile phone. Users rate and tag photos. This information is then used to create a photobook.

large, failure-prone software projects into smaller tasks with corresponding time estimates. We propose the concept of *selfsourcing*, in which personal tasks are broken down into microtasks to be completed by the individual. By algorithmically providing structure to a task and supporting context maintenance, selfsourcing enables users to complete complex tasks in short bursts of time via very small work items.

Selfsourcing builds on crowdsourcing, where small tasks are completed by a large group of remote workers. While often used for simple tasks, crowd work is increasingly being composed to accomplish complex tasks that are not obviously achievable via microtasks [8], such as taxonomy creation [2]. Complex personal tasks that require deep personal knowledge or contain private information [11] can likewise be accomplished in small steps by an individual using the same processes. Selfsourcing allows individuals to leverage crowdsourcing's task decomposition [8] and context maintenance techniques [10] to improve their ability to complete tasks that cannot be done by others.

Selfsourced microtasks are ideal for inserting into the interrupted time between larger tasks. Information workers often find themselves interrupted [12], and research suggests that it is hard to resume a task when this happens [3], taking up to 15 minutes to return to focused activity [7]. Given the average information worker is interrupted faster than it takes to achieve full efficiency [12], interruptions cause a significant loss in productivity [3]. However, resumption is easier when a person is interrupted at a breakpoint [6], and when the task being returned to has a clearly achievable short-term outcome [13]. Researchers have tried to use these insights to decrease interruption costs by strategically scheduling interruptions to occur at breakpoints [6], helping users set goals upon interruption [13], and reminding users of their goal upon return [3]. In contrast, selfsourcing changes the nature of the task itself to make it more interruption-friendly. Large, overwhelming tasks are transformed into small, achievable components, facilitating recovery from interruptions, benefiting overall productivity, and helping people take advantage of time that might otherwise be wasted.

Selfsourcing Examples

We applied selfsourcing to two personal information tasks, photo organization and brainstorming. In both cases, existing approaches to task decomposition allowed us to convert these otherwise overwhelming tasks into manageable microtasks. Users can then complete the microtasks via short bursts of activity to create a significant end product. These examples illustrate selfsourcing's potential benefits and challenges.

Photos → Photobook

People have amassed large archives of digital photographs that have significant value to them but are difficult to use because they lack structure. We developed a mobile selfsourcing application that supports the creation of a photobook from an unorganized set of photographs (see Figure 1). The application runs on Windows Phone and works with photo collections stored in the cloud. Users perform five types of microtasks in the following order:

Ignore: Photograph collections often contain a number of unimportant pictures, including pictures taken accidentally (e.g., of the inside of one's pocket) or for administrative purposes (e.g., of a receipt). Users identify these unimportant photos to ignore by selecting their thumbnails from a set of temporally grouped pictures.

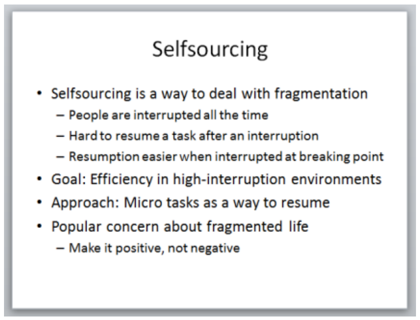

Figure 2. A slide from a presentation on selfsourcing created via microtasks. The slide author brainstormed ideas, extended those ideas, and organized them using selfsourcing.

Rate: In addition to ignoring useless photos, it is important to identify particularly good ones. Users are asked to rate pictures that have not been ignored on a scale of one to five stars (Figure 1, top).

Tag: Users are also asked to tag photographs with associated names of people, places, and events. The app collects tags for the highest rated photographs, and applies these tags to other photos using Tag Extend.

Tag Extend: To extend a tag to additional photos, users are shown an existing tag and thumbnails that are similar to an already tagged photo (Figure 1, bottom). They are asked to select the photos to which the tag applies. Photos are 'similar' if they were taken close in time, or contain similar faces, locations, or image features.

Compare: Photos with the same tag may have the same rating, making it hard to identify the best photo with that tag. Users are thus asked to choose their favorite picture among two with similar ratings and tags.

These microtasks are used to generate a photobook from the user's collection by selecting the best photos for each tag, sorting them by time, and using an online photobook creator to lay the pictures out in a book.

Brainstorming → Presentation

We also implemented a desktop selfsourcing application to support brainstorming and create a presentation with the results. A slide from a selfsourced presentation (on the topic of selfsourcing) is shown in Figure 2. Brainstorm-based presentations are created in three phases:

Idea Generation: During the brainstorming phase a user is asked to enter short fragments of ideas on a topic. The user is also prompted with previously entered ideas and asked to riff on them [4]. For example, the presentation in Figure 2 was created using 48 unprompted ideas and 148 ideas inspired by the initial 48.

Idea Organization: Once a set of ideas has been collected, these are organized using a modification of the Cascade taxonomy creation [2]. The user takes up to three passes to tag each idea, adding one or more tags at a time. A third is pass used only if the tags for the first and second pass do not match. This resulted in 61 unique tags for the example in Figure 2. Uncommon tags are automatically removed, and tags that are highly correlated are suggested to the user to merge.

Presentation Creation: Finally, the tagged ideas are used to automatically create a presentation. Individual slides are constructed iteratively. The tag with the fewest ideas associated with it is used as the slide header, and the associated ideas make up the bullet points. Since ideas can have multiple tags, subgroups within a slide are made using auxiliary tags. The ideas included in the slide are then removed from the general pool, and the tag with the next fewest number of ideas is considered. This ensures that individual slides do not contain too many ideas. Slides are ordered so that those associated with tags that cover the most ideas appear first. These slides only contain the ideas that remain associated with the tag after the ideas related to other tags are removed, and tend to address the big concepts that emerged from brainstorming. The user then copyedits slides and performs local reorganization.

Discussion of Selfsourcing

Using photo organization and brainstorming as examples, we now discuss issues related to selfsourcing that are worthy of further study, including task breakdown and allocation, user motivation, and collaboration.

Selfsourcing Task Breakdown

We begin by looking at the design of selfsourcing tasks. Task structure can be provided up front, or developed dynamically. For example, TaskGenies use crowd workers to create action plans with concrete steps that help individuals successfully complete personal tasks [9].

While crowdsourcing provides insight into how to decompose tasks, there may be reasons to decompose selfsourced tasks differently. Selfsourcing microtasks, for example, may be larger than those typically sent to the crowd. In the case of presentation creation, users review the entire presentation after creation to ensure proper flow. In this way, selfsourcing lets them complete the little tasks that need to be done to create a presentation in small fragments of time, while focusing on the big picture during the larger units of time. Additionally, while significant work in crowdsourcing has focused on quality assurance, selfsourcing requires less validation because the person performing the work has a vested interest in the task being performed well. For example, while our photobook and presentation creation use the approaches described by Chilton et al. [2], they require fewer rounds and no validation.

Selfsourcing Task Allocation

The most common approach for allocating crowdsourcing tasks is for a worker to select one to perform from among many. Selfsourcing applications could allow users to select their own tasks, or have the next task algorithmically determined. In our photobook example we use a dependency graph with some random variability to automatically allocate tasks. However, some tasks (e.g., ignore) are easier than others (e.g., tag), and could be allocated according to the user's available cognitive resources. Users could triage tasks as part of the selfsourcing process, but this creates additional work for the user and is difficult to do without a global view of task progression. Alternatively, users could specify their constraints (e.g., available time and attention) at the start of a session and the system could use this information to algorithmically provide the most appropriate tasks. The system could then learn to predict these constraints over time. The user's device also serves as a constraint. Tagging ideas for a presentation or photographs for a photobook may be easy to do from a mobile device, but extended text input for brainstorming and viewing of a final product like a presentation is best done on a desktop computer.

When a complex task is broken down, it can be important to instantiate context for the component task. Researchers have explored how to support the generation and use of context in crowdsourcing [10]. Different from crowdsourcing, individuals performing selfsourced tasks have significant long-term context (since the task belongs to that person) and short-term context (since sets of tasks may be performed in sequence). The allocation of selfsourcing tasks can be used to support context development. At the start of a session, users can be given tasks that do not require context, and progress to tasks that require more context as the session progresses. For example, a person organizing their photos may initially be asked to tag photos, and then be asked to extend those tags while their meanings are still fresh in their mind. Conversely, reduced context can sometimes be beneficial. Variation among crowd workers is considered positive in that it introduces multiple perspectives. Selfsourcing tasks must actively seek variation, by, for example, requesting tags for the same item at different times [14].

Motivation for Selfsourcing

Crowd workers have extrinsic incentives to motivate them (typically financial), while people performing their own tasks must be intrinsically motivated. Task decomposition can provide additional motivation [9]. People find it hard to estimate task workload and length in order to schedule time to complete the task in a single, uninterrupted session because they have few reference points or schemas to compare against [5]. Selfsourcing removes this uncertainty by helping users efficiently schematize their work. People's motivation to complete selfsourcing tasks can further be enhanced by making microtasks enjoyable. In the case of photo organization, for example, people take pleasure in viewing personal photographs that they like. For this reason, users are asked to tag photographs they rate highly, and merely extend those tags to lower rated photographs. Periodically changing the type of task a user is asked to perform could also help keep people engaged.

Users can also be motivated by seeing their accomplishments. Because microtasks are measurable, progress towards completion can be quantified and reflected back to the user. A person may be willing, for example, to rate a few additional photographs if they know that doing so will complete a task milestone. People may also be motivated by seeing their final output develop and improve. In our examples, a rough presentation or photobook can be constructed after the user has completed only a portion of the necessary microtasks. Additional input refines the output, but intermediate outputs can provide a sense of accomplishment. Users may choose to use selfsourced microtasks to engage initially with a task they find hard to start, and then move to working on the final product once they become engrossed in the task.

Because the information the user provides during selfsourcing is schematized, selfsourcing systems can also use the same input to create different outputs. It is much easier to build new outputs from microtasks than change one end product into another. For example, in addition to creating a photobook, the photo tasks could be used to select photos to post to Facebook. Such intermediate outputs could serve as short-term goals.

Selfsourcing as a Way to Share Work

While selfsourcing tasks can be completed by an individual, the approach makes it easy to share aspects of a task with others in a way that is not easy to do for traditional complex tasks. For example, a spouse can help create a photobook by tagging photos, and colleagues can support presentation creation by adding ideas during brainstorming. This not only reduces the amount of work for the user, but also provides new perspectives [14]. For example, brainstorming works particularly well when individuals build off each other's ideas [4]. Some selfsourced microtasks can be sent to friends (*friendsourced* [1]) or crowdsourced. For example, a user may not want to rate all of their photographs, but would be willing to pay the crowd to help.

The decomposition of personal tasks in selfsourcing also makes it easy to incorporate automation. For example, our photo application uses face recognition to help users propagate tags to other photographs. The user and the system collaborate by having the system make an algorithmic guess, and then having the user sanity check that guess through the Tag Extend task. Interactive learning fits well into the selfsourcing framework, and microtasks may be particularly easy to learn because they are schematized. Selfsourcing provides a straightforward way for users to trade off factors such as costs, effort, tolerance for errors, and privacy, when soliciting input from other sources.

Conclusion

We have proposed the concept of selfsourcing, wherein personal tasks are broken into microtasks and completed by the individual themself, much in the same way that crowdsourcing breaks tasks down to be completed by crowd workers. We believe that it is possible to decompose many existing information tasks into manageable chunks via selfsourcing to create an enjoyable and productive user experience. Using photo organization and presentation creation as examples, we discussed a number of interesting issues and opportunities raised by selfsourcing, including the design and allocation of tasks, its impact on motivation, and how selfsourcing facilitates the sharing of personal information work.

Selfsourcing is meant to complement, not replace, deep big-picture thinking. Modern information tasks take place in the face of many interruptions [7, 12], and we must adjust how we complete our personal tasks accordingly. While many selfsourced microtasks require focus on small details, the approach allows people to accomplish them in short bursts of activity, potentially with the help of other people or algorithms, and use their focused periods of cognitive efforts on the key high-level aspects where they can contribute the most.

References

[1] Bernstein, M.S., Tan, D., Smith, G., Czerwinski, M. & Horvitz, E. Personalization via friendsourcing. *TOCHI* 17(2), 2010.

[2] Chilton, L.B., Little, G., Edge, D., Weld, D.S. & Landay, J.A. 2013. Cascade: Crowdsourcing taxonomy creation. *CHI 2013*.

[3] Cutrell, E., Czerwinski, M. & Horvitz, E. Notification, disruption, and memory: Effects of messaging interruptions on memory and performance. *Interact 2001.*

[4] Faste, H., Rachmel, N., Essary, R. & Sheehan, E. Brainstorm, chainstorm, cheatstorm, tweetstorm: New ideation strategies for distributed HCI design. *CHI 2013.*

[5] Hart, S.G. & Staveland, L.E. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Advances in Psychology,* 52, 1988.

[6] Iqbal, S. & Bailey, B. Effects of intelligent notification management on users and their tasks. *CHI 2008*.

[7] Iqbal, S. & Horvitz, E. Disruption and recovery of computing tasks: Field study, analysis, and directions. *CHI 2007*.

[8] Kittur, A., Nickerson, J.V., Bernstein, M.S., Gerber, E.M., Shaw, A., Zimmerman, J., Lease, M. & Horton, J.J. The future of crowd work. *CSCW 2013*.

[9] Kokkalis, N., Köhn, T. Huebner, J., Lee, M., Schulze, F. & Klemmer, S.R. TaskGenies: Automatically Providing Action Plans Helps People Complete Tasks. *TOCHI,* 20(5), 2013.

[10] Lasecki, W.S., Murray, K.I., White, S., Miller, R.C. & Bigham, J.P. 2011. Real-time crowd control of existing interfaces. *UIST 2011*.

[11] Lasecki, W.S., Teevan, J. & Kamar, E. Information extraction and manipulation threats in crowd-powered systems. *CSCW 2014*.

[12] Mark, G., Gonzalez, V. & Harris, J. No task left behind? Examining the nature of fragmented work. *CHI 2005*.

[13] Trafton, J.G., Altmann, E.M., Brock, D.P. & Mintz, F.E. Preparing to resume an interrupted task: Effects of prospective goal encoding and retrospective rehearsal. *IJHCS* 58, 2003.

[14] Vul, E. & Pashler, H. Measuring the Crowd Within: Probabilistic Representations within Individuals. *Pyschological Science*, 19(7), 2008.