Web Workers Unite! Addressing Challenges of Online Laborers

Benjamin B. Bederson
Human-Computer Interaction Lab
Computer Science Dept.
University of Maryland
College Park, MD 20742
bederson@cs.umd.edu

Alexander J. Quinn
Human-Computer Interaction Lab
Computer Science Dept.
University of Maryland
College Park, MD 20742
aq@cs.umd.edu

CHI 2011, May 7–12, 2011, Vancouver, BC, Canada.
ACM 978-1-4503-0268-5/11/05.

Abstract
The ongoing rise of human computation as a means of solving computational problems has created an environment where human workers are often regarded as nameless, faceless computational resources. Some people have begun to think of online tasks as a "remote person call". In this paper, we summarize ethical and practical labor issues surrounding online labor, and offer a set of guidelines for designing and using online labor in ways that support more positive relationships between workers and requesters, so that both can gain the most benefit from the interaction.

Keywords
Human Computation, HCOMP, collective intelligence, crowdsourcing, wisdom of crowds.

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

Introduction
The recent growth in the fields of "human computation" [12] and crowdsourcing [8] has brought with it expanded interest in online labor. Popularized by Amazon Mechanical Turk (AMT – www.mturk.com), and now available through other services such as

CrowdFlower (www.crowdflower.com) and CloudCrowd (www.cloudcrowd.com), online labor is a practice in which *requesters* post tasks to a public web site where people do the tasks in exchange for money. Workers and requesters often remain anonymous. In addition to human computation and crowdsourcing, online labor has also been an enabling platform for experiments and practical application of collective intelligence [10] and wisdom of crowds [16].

There are, of course, many kinds of online labor representing varying approaches, levels of skills, work domains, etc. Dozens of sites, such as www.topcoder.com, www.99designs.com, and www.innocentive.com, support such markets. These are all relevant, but this paper focuses on the narrower, lower skilled, and more general marketplaces such as AMT.

By performing tasks on demand from around the world, these *web workers* offer an inexpensive and high throughput solution to a variety of problems that only humans can solve. Web workers have the potential to earn money in places where they may otherwise have poor job prospects. Unfortunately, as with any labor market, once humans and money are involved, a host of problems surface. From privacy breaches to unpaid or underpaid labor, there are real social risks that arise with the use of these technologies [5, 14].

While these social risks may at first glance seem beyond the scope of HCI researchers (and particularly technologists), we feel that it is precisely our responsibility to address them from the beginning since it is designers, not ethicists or policy makers, who have the power to influence what is built and to mitigate risks before any harm is done. This is especially important because technologies tend to build on each other and so once things are released, it can be difficult or impossible to reverse the social effects [6, 9].

As we will discuss below, much of the challenges posed by this kind of work can be attributed to the anonymity of all parties, unchecked authority of the requester to decide payment terms, and the general imbalance of information.

Our goals with this paper are to:

- Summarize the ethical concerns about online labor.
- Explore some issues related to fair wages.
- Describe some of our own experience.
- Suggest strategies to help mitigate these problems.

Problems for web workers

Only a handful of research papers have examined the problems web workers face [5, 13, 14, 15]. Silberman et al. [14, 15] looks into the concerns web workers themselves report. The biggest set of issues raised by all centers on wages. There are numerous concerns related to unfairly rejected work, slow payment, and payments that do not fairly reflect the work performed for the given task. Furthermore, there are occasional technical problems, the cost of which is typically borne by the workers. When such problems arise, workers often complain that requesters are unresponsive to emails (which are initially mediated by AMT to maintain anonymity).

This concern about wages is reflected in comments posted at Turker Nation [17], a web-based forum where AMT workers (often called “Turkers”) are free to discuss their situations openly. They describe their concerns, often emphatically. Issues may be discussed in general terms or by naming specific requesters and problems encountered. The site contains a user-maintained blacklist of requesters to avoid. The other major resource is Turkopticon [18], a plug-in for the Firefox web browser that lets workers rate the credibility of requesters. Turkopticon annotates each HIT with the collective rating of the requester, based on communicativity, fairness, generosity, and promptness.

The fact that these grassroots efforts exist and are used is interesting because it demonstrates that at least some web workers do not feel that they are adequately supported by the existing infrastructure. At the same time, it is equally clear that many workers behave badly toward requesters, expending no more than the minimum effort necessary to receive payment for doing a HIT [3, 11].

This behavior is not surprising given the theory of moral hazard [7], which says that when one party does not incur the full cost of their actions, they tend to behave more riskily, and thus increase the cost of the other involved parties. Given the anonymity of all participants in AMT, moral hazard is in full play. Because requesters face few consequences, they can and do post unethical or even illegal tasks, or deny payment for work completed. Similarly, workers may do substandard work or more blatantly cheat. Cheating often takes the form of clicking or typing randomly, using scripts to enter useless input, or using online resources to find answers that are not useful to the requester, but just relevant enough to bring about payment (i.e. using an online translation engine to create a translation into a language the Turker does not know).

These kinds of behaviors, while obviously detrimental to the party who bears the costs, also have significant downsides for the market as a whole. The *market for lemons* theory models what happens when a buyer cannot accurately judge the quality of an individual product prior to committing to its purchase, leading the buyer to average the quality of all similar products in their decision as to how much to pay [1]. This creates an incentive for sellers to lower the quality of their goods, since they will be paid an average price in any case, and thus they can benefit more from each transaction if the payment they receive is greater than the value of what they gave in return. Good workers, on the other hand, tend to leave the market because they get paid less than their actual value. The result is that the market decreases in overall quality, resulting in a market that sells only poor quality items, i.e., lemons.

It might seem that this theory would not apply to AMT since requesters can choose not to pay for poor work. However, the nature of online labor is that it is difficult to automatically judge the quality of the results, since the work was too difficult for computers to do in the first place. Furthermore, payment for most tasks is a flat rate per task, leading many workers to do work that is just “good enough”. Therefore, the requesters’ ability to deny payment does not adequately mitigate the factors that lead to the market for lemons environment. The resulting environment is somewhat like a market for lemons where the requesters are like buyers without knowledge of the product, the workers are like sellers who have an incentive to do their work poorly, and the work done is, unfortunately, all too frequently a lemon.

Requesters have developed sophisticated techniques to mitigate these problems by, for example, using other web workers to judge the quality of a first round of work [12]. When used effectively, these measures add significantly to the cost of getting work done, thereby reducing the value of online labor markets in the first place. Other solution providers, such as CrowdFlower and CloudCrowd, have systems in place to automatically identify workers who submit poor quality work. This is typically done using some gold standard data and worker reputations. However, mechanisms that are general necessarily add to the base cost of getting the work done.

Another area of concern, raised by Jonathan Zittrain, is *moral valence* [19], which supposes that anonymity extends *beyond* the worker and the requester, all the way to the nature and purpose of the work itself. There are numerous tasks that are clearly objectionable (e.g., requests for posting positive reviews about products the worker is not actually familiar with). However, some tasks extract small pieces of larger jobs and present them to workers out of context so that workers do not understand the purpose of the work they are doing. This is more challenging because the work might be objectionable if the worker understood its context. For example, Zittrain gives the example of matching photos of people, which is a task that sometimes comes up today in HITs. That same task could be used by an oppressive regime to identify citizens who have, for example, participated in a peaceful political demonstration.

Wage Issues for Web Workers

Wage issues have been the largest source of complaints raised by web workers [14, 15]. They say requesters sometimes pay late, pay too little, or don't pay at all. Before a HIT can be created, the requester must add sufficient funds to the account with Amazon to cover all of the work. However, the funds are frozen and not paid to workers until the requester has confirmed that the work was satisfactory. To cover the possibility that the requester might neglect to approve the work, requesters are required to specify an "auto-approval delay" after which funds will be released to workers. The maximum is thirty days. We consider it polite to set the delay to three days, or a week at most, and we regularly receive e-mail from workers after just one or two days if payment has not yet been approved. This indicates how the expectations of web workers match the speed of the web, and also perhaps the lack of trust between web workers and requesters. Since the auto-approval delay is not visible to workers, they have no visible guarantee of when they will be paid.

Given that web workers also complain that some requesters unfairly and arbitrarily reject good work, this lack of trust is perhaps not surprising. AMT is set up so that the requester has unchecked power over this important judgment. There is no rebuttal process, and if a worker is not paid by the requester, the only recourse is to avoid the offending requester in the future and voice their concerns using forums such as Turkopticon or Turker Nation.

Early on, our own strategy to deal with quality issues was to seed the job with ground truth, and reject all work from workers whose overall score fell below a baseline level. Of course, there is always the possibility that such a good worker might honestly misunderstand the instructions. Therefore, we would send a warning after ten wrong answers and block the worker after twenty. At first, this seemed a good strategy, but we later learned that blocking workers and/or rejecting a lot of tasks at once can lead to the worker being permanently banned by Amazon. In creating schemes for mitigating dishonest work, we need to consider ethical implications with respect to the workers and fair treatment should be the default.

Unfortunately, the consequences of such problems can be very real for workers, especially if they depend on the money. For example, we recently mistakenly rejected some work done by several workers. One of the workers, who had done 248 of our tasks (the most of any worker in that experiment) was subsequently banned by Mechanical Turk. The worker, who is in India, said he lost the entire $130 that was in his Amazon payment account. Despite prodding by us and the worker, Amazon refused to reverse the action. Their reply to the worker (which the worker forwarded to us) refused any explanation, stating only that the account had been "…closed due to a violation of [the] Participation Agreement and cannot be reopened" and that "Any funds that were remaining on the account are forfeited, and we will not be able to provide any additional insight or action." Our account screens confirmed that our payments to that worker were being withheld. In this case, we could have designed our system to only reject work after some automated cross-checking for errors and a case-by-case human evaluation.

This case also offers a lesson about anonymity. The designers of Mechanical Turk chose to hide the identity of workers from the requesters who post tasks. Typically, we receive non-anonymous email from real human workers only if we have made a mistake in conducting our experiments. When we were designing this experiment, we could regard the workers as a computational resource. Only after there was a problem did the relationship become more personal. The lesson is that although these systems place a degree of separation between us and the workers, in essence it is still an employment relationship.

Despite the problems, one can imagine why AMT may have been structured in this way. If workers had greater power to demand redress, they could always claim that their work was performed legitimately, even if it was abundantly clear that it was not. For example, in a recent experiment that we ran, we posted Chinese linguistic tasks at $0.05 each and offered a $0.05 bonus per task for workers who completed at least 100 tasks within the final 28 hours. Of the 8,498 responses received, 3,174 were submitted fraudulently by a single worker, apparently using a script.

Similarly, requesters can be malicious as well. There are certainly HITs available asking workers to do dubious work (i.e., providing high ratings at e-stores). Not only might those requesters not pay the workers, but if the worker's real identities were available, then the requester might sell their contact information to spammers, etc. So anonymity and identity issues remain a significant ongoing challenge.

Effective Hourly Wage

The issue with perhaps the most subtleties is the most basic: How much should workers be paid for their work? Assuming the work is performed honestly, it is still difficult to judge how much web workers typically make. There clearly is significant variation, and the actual hourly wage of any individual worker depends on many factors which go beyond the obvious price per task and the time it takes to do the task. For example, some tasks have a significant learning curve. Thus, workers can only justify the time spent understanding the task if there are many instances.

While an hourly wage is difficult to predict, it can still be estimated. But is it a good idea to offer a payment system based on time spent rather than work done? In the physical world, both models are commonly supported with both regular and temporary employees frequently paid by the hour alongside of contract workers paid by the job. Payment incentives and wage structures result in complex human behavior. Even basic worker strategies, such as maximizing hourly wage are not consistently followed.

For example, a study of taxi drivers showed that many of them employed the strategy of working as long as necessary each day to reach a certain predetermined amount of earnings [4]. A consequence was that they worked longer hours on bad days (on which it is harder to find passengers) than on good days - exactly the opposite of the pattern that one would expect if they were trying to maximize their hourly earnings. Apparently, driving each day until you have collected a certain amount of money is cognitively simpler than trying to maximize your hourly earnings under changing circumstances.

Given the complexity of real life decision making, coupled with the challenges of monitoring how web workers spend their time while working, it is easy to understand why requesters might be reluctant to offer anything like an hourly wage. Even so, hourly wage remains a hot topic of discussion as represented by a post by Luis von Ahn [2]. The fundamental issue is whether web workers should be paid based on their time spent working.

Given the current structure of online labor, this may seem like an impractical idea. And, after all, in the physical work, there is plenty of labor that is paid for as “piece work”, i.e., by the actual amount of work done. On the other hand, in the physical world, hourly payment with a minimum wage is not only common, but required by law in numerous situations in the vast majority of countries in the world. So, we believe it is at least worth considering the characteristics of minimum wage.

The fundamental motivation for a minimum wage is that it might decrease poverty by ensuring that workers earn enough to survive. There is also the moral argument that workers should not be taken advantage of by offering an unlivable wage when the employer knows that the worker has no other options. There are also practical arguments such as the idea that minimum wage is a better way to help the poor than welfare because it requires that people actually do something for their government assistance. Higher wages also have the potential of attracting higher quality workers, which could result in a virtuous cycle, improving the relevant industries.

However, there are a number of arguments against a minimum wage. The biggest issue is that minimum wage may not, in fact, succeed in decreasing poverty because while some workers get a higher wage, others may lose their jobs altogether since some employers won't be able to pay that higher wage. There also is a concern related to the "market for lemons" which is that guaranteeing a minimum wage would overpay poor quality workers, thus disincenting good workers from doing their job well.

Web worker characteristics bring a number of additional concerns about minimum wage. First of all, the argument that monopolistic employers can abuse workers because they offer the only employment in a region largely goes away since the decentralized nature of online labor increases worker options. An important practical issue is that minimum wages are tied to the local standards of living where workers live. While it would be possible to establish different minimum wages for different locales, this would be problematic since online labor is a true global marketplace. Such an approach would direct work to locales with the lowest minimum wage, leaving out workers in areas with higher minimum wages. Finally, there is the practical issue that it is impossible to reliably determine how long a web worker actually works on a given task. Similarly, it is not even possible to reliably know where a given worker is located, since workers may be connected to the internet via proxy servers in other countries (i.e., to bypass government internet filters).

These issues make a conventional minimum hourly wage for online labor likely to be impossible. Nevertheless, web workers' concerns provide enough motivation for us to at least consider issues of hourly wages.

We think that a good starting place is for requesters to calculate the expected hourly wage of workers doing their tasks—through a combination of doing some of the tasks in-house, as well as monitoring the time spent by (non-cheating) web workers. Even if requesters choose to offer the lowest wage possible, they could still post the information they do have (i.e., "est. $1.00 per hour"). A system such as AMT could make this the norm by posting the estimated hourly wage observed with the work done so far on each task. However, even this approach would overestimate time spent since time would be counted when, for example, a worker switches to another task for a short while.

Design Guidelines

Any system can be abused, but we think that a system's design can encourage less problematic use. To that end, we make the following recommendations for requesters based on our experience with online labor markets. We think that if implemented, it would result in more ethical utilization of this labor source that could result in better economics with higher quality and more efficient work, benefiting all parties.

Requester design guidelines

1. **Hourly pay:** Price tasks based on time. The time to do tasks can be estimated in-house before posting HITs.
2. **Pay disclosure:** Disclose the expected hourly wage.
3. **Value worker's time:** Optimize tasks to use worker's time effectively. We have seen many HITs on AMT that were poorly designed, forcing workers to spend excessive time navigating the task interface, waiting for network activity, or dealing with technical problems.
4. **Objective quality metrics:** Decide to approve or reject work based on objective metrics that have been defined in advance and disclosed to workers.

5. **Immediate Quality feedback:** Give immediate feedback to workers, showing whatever metrics are available. This technique is exemplified by CrowdFlower's system, which displays a panel above the task with the worker's estimated accuracy so far.

6. **Longer-term feedback:** Give warnings to problematic workers. For example, our system sends email to workers who submit 10 HITs below a quality threshold. This gives them the opportunity to write to us and ask for clarification if they have misunderstood something about the instructions.

7. **Disclose payment terms:** Disclose in advance when payment will be made. Turkers have indicated to us that 24-48 hours is the maximum acceptable delay between submitting work and receiving payment.

8. **Follow payment terms:** Pay as promptly as possible, and always within the disclosed timeframe. Ideally, the service (such as AMT) would more explicitly disclose the payment timeframe for all HITs.

9. **Provide task context:** Given the risk of doing objectionable work, HITs should be described in the context of why the work is being done. Not only is this likely to make the work more satisfying to workers, but it lets workers make informed ethical choices about how they spend their time. Even if you do not foresee any ethical objections to your task, stating the context could potentially help establish a norm, so that tasks that do not disclose the context will stand out from the rest.

System design guidelines

1. **Limit anonymity:** The mutually anonymous nature of most current online labor systems is a significant driver of many of the issues raised in this paper. Anonymity of requesters enables them to reject good work with near impunity. It also enables them to post unethical or illegal tasks with no public scrutiny. Anonymity for workers enables them to cheat at large scales with nearly no risk since, as with requesters, if their reputation gets damaged, they can simply create a new account.

   Online labor marketplaces should make it more difficult for someone to create multiple accounts. Even if they hid the identification of the requester and worker, they could require positive and unique identification per account. However, we think if the actual identity of requester and worker alike were available, it would lead to fairer labor practices.

   Furthermore, showing real identities could add significant value if non-anonymous participants could create an online portfolio showcasing their work and skills. This could improve their chance of getting future work (both within and external to the system). A great example of this kind of complementarity can be found in www.stackoverflow.com, a programmer's Q&A site.

2. **Provide grievance process:** Provide a fair means for workers to request a review of work that was rejected. Having the requester maintain complete control over the payment of work creates an imbalance of power. At the very least, requesters should explain by email to workers exactly why their work was rejected and what they would have had to do in order to get their work accepted.

   Ideally, a community-run arbitration system could be set up when simple negotiation is not acceptable to the worker. While such an arbitration system would not be binding, the results could be publicized and included in a simple service such as Turkopticon.

Conclusion

Significant further work remains in both understanding and improving online marketplaces. From processes to better estimate the time needed for tasks to creating ways of increasing transparency on all sides while protecting requesters and workers alike, we clearly are at the forefront of this frontier.

We remain excited and optimistic about online labor, and believe it will continue to grow and become a significant part of the global intellectual labor pool. In order to realize the potential of this form of labor, practices must be modified and regularly considered to ensure the fair and equitable treatment of requesters and workers alike.

References

1. Akerlof, G.A. The market for "lemons": Quality uncertainty and the market mechanism. The quarterly journal of economics, 1970.
2. Ahn, L.V. “Work and the Internet”, blog post 7/26/2010. http://vonahn.blogspot.com/2010/07/work-and-internet.html
3. Bederson, B.B., Hu, C., & Resnik, P. (2010) Translation by Interactive Collaboration between Monolingual Users, Proceedings of Graphics Interface (GI 2010), 39-46.
4. Camerer, C., Babcock, L., Loewenstein, G., & Thaler, R. (1997). The Quarterly Journal of Economics, 112 (2), May, 1997, pp. 407-441.
5. Felstiner, A. and Hall, B. 2010 Working the Crowd: Employment and Labor Law in the Crowdsourcing Industry. http://ssrn.com/abstract=1593853.
6. Friedman, B. Value-sensitive design. Interactions. 1996.
7. Hölmstrom, B. Moral hazard and observability. The Bell Journal of Economics, 1979.
8. Howe, J. Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business, Crown Publishing Group, New York, 2008.
9. Lanier, J. 2010 You Are Not A Gadget: A Manifesto. 1st. Knopf Publishing Group.
10. Malone, T.W., Laubacher, R., and Dellarocas, C. Harnessing Crowds: Mapping the Genome of Collective Intelligence. MIT Sloan Paper #4732-09, 2009.
11. Resnik, P., Buzek, O., Hu, C., Kronrod, Y., Quinn, A., & Bederson, B.B. (2010) Improving Translation via Targeted Paraphrasing. EMNLP 2010. (in press)
12. Quinn, A. J., Bederson, B. B. 2011. Human Computation: A Survey and Taxonomy of a Growing Field. In Proceedings of CHI 2011, ACM. (in press).
13. Ross, J., Irani, L., Silberman, M. S., Zaldivar, A., and Tomlinson, B. 2010. Who are the crowdworkers?: shifting demographics in Mechanical Turk. In Proceedings of CHI 2010, Atlanta GA, ACM, 2010.
14. Silberman, M. S., Ross, J., Irani, L., Tomlinson, B. Sellers' problems in human computation markets. In SIGKDD HCOMP 2010, Washington, DC, July 2010.
15. Silberman, M.S., Irani, L. and Ross, R. 2010. Ethics and tactics of professional crowdwork. *XRDS* 17, 2 (December 2010), 39-43. http://doi.acm.org/10.1145/1869086.1869100
16. Surowiecki, J. The Wisdom of Crowds. Anchor Books, New York, 2005.
17. Turker Nation www.turkernation.com
18. Turkopticon. http://turkopticon.differenceengines.com
19. Zittrain, J. The Future of the Internet And How To Stop It. Yale University Press, 2009.