

Platform Design for Crowdsourcing and Future of Work

David Gross-Amblard
University of Rennes, France
dga@irisa.fr

Atsuyuki Morishima
University of Tsukuba
mori@slis.tsukuba.ac.jp

Saravanan Thirumuruganathan
QCRI, HBKU
sthirumuruganathan@hbku.edu.qa

Marion Tommasi
University of Lille and INRIA
marion.tommasi@inria.fr

Ko Yoshida
CrowdWorks
yoshida@crowdworks.co.jp

Abstract

Online job platforms have proliferated in the last few years. We anticipate a future where there exists thousands of such platforms covering wide swathes of tasks. These include crowdsourcing platforms such as Amazon Mechanical Turk (AMT), CrowdWorks, Figure Eight; specialized services such as ride-hailing; matching markets such as TaskRabbit that matches workers with local demand and so on. It is widely anticipated that a vast majority of human workforce will be employed in these platforms. In this article, we initiate discussions about the under studied aspect of platform design – how to design platforms that maximize the satisfaction of various stakeholders. We also contribute a novel taxonomy for platform ecosystems that categorizes existing and emerging platforms. Finally, we discuss the need for interoperability between these platforms so that workers and requesters are not tied to a single platform.

1 Introduction

Online job platforms are a class of alternative job arrangements that have widely proliferated in the last few years. These could be general purpose platforms such as Amazon Mechanical Turk or CrowdWorks that support a wide variety of tasks entirely online. Alternatively, they could be specialized platforms such as for ride hailing services where all tasks are of the same type. Hybrid platforms such as TaskRabbit matches *labor* providers such as plumbers with requestors who need such services. Even platforms such as Etsy could be construed as a job platform as it allows *goods* providers to sell their products. Job platforms acts as an intermediary by matching workers with requestors while taking a cut in the payments. As much as 36% of US workers have an alternative work arrangement in some capacity [8]. This number is steadily increasing and it is in the realm of possibility that the vast majority of human workforce will be employed in such arrangements in the near future.

We anticipate a future where there are thousands of online job platforms for different types of tasks. Due to their bespoke nature, this will not be a winner takes all. Despite the impending Cambrian explosion of such platforms, there has been very limited work on *platform design* – what are the current pain points of current platforms and how can they be better designed to handle the future of work? In this article, we investigate this important problem and make a number of concrete suggestions.

Copyright 0000 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

There are three major stakeholders – workers, requestors and the online job platforms. It is important that the platform design takes into account the requirements of each of them. The current generation of platforms are grappling with how to satisfy each of them. So far, the efforts are ad-hoc and error prone wherein none of them are satisfied with the status quo.

Current generation of platforms support simple microtasks only. Support for complex tasks is often rudimentary at best. We believe that in the future, platforms must be able to support complex knowledge intensive and collaborative tasks that could involve sophisticated workflows. When there are a large number of job platforms, it is very important to have formally defined mechanisms to allow platforms to inter-operate with each other. This is often beneficial for workers (they could work on multiple platforms at once) and for requestors (recruit workers from multiple platforms). A worker should be able to move between platforms without losing any of the skills/ratings from the original platform. Similarly, a requester must be able to post a task where workers from multiple platforms are able to contribute. Achieving this requires tackling a number of technical and policy challenges. The technical challenges include agreeing upon a predefined schema and API interfaces. Furthermore, one must identify a class of queries that allows exchange of information about workers, requestors and tasks. The regulatory policies must ensure that platforms play well together.

Paper Outline. The rest of the paper is organized as follows. In Section 2, we conduct an user study over multiple popular crowdsourcing platforms whose results indicate the need for platform design. In Section 3, we enumerate a number of key functionalities that are either missing or not widely supported in the current generation of crowdsourcing platforms. However, these are also the functionalities that were desired by stakeholders or have the potential to improve their satisfaction. In Section 4, we discuss the important problem of platform interoperability. As more and more online job platforms are created, it is important that they can interact with each other and we identify the key dimensions for enabling this functionality.

2 Motivation

Crowdsourcing platforms has been extremely well studied by different communities [3]. We believe that they are harbingers of the oncoming shift towards the gig economy brought upon by online job platforms. By studying how workers fare in these platforms allows us to extrapolate the findings to Future of Work (FoW).

Crowdsourcing platforms including Amazon Mechanical Turk (AMT) and CrowdWorks have a number of similarities to online job platforms. Requestors create various tasks that include information such as description of the work required, compensation provided, requestor details etc. The requestor can also filter workers based on previous experience. When a worker logs on, she could see all the available tasks and choose to work on a subset of them. When the worker submits a task, it could be reviewed by the requestor. If deemed satisfactory, the worker is paid. If not, the worker submission is rejected. The crowdsourcing platform takes a cut from the payment made by requestor to the worker. This basic model pioneered by AMT has become ubiquitous in online job platforms as diverse as Uber, TaskRabbit and so on.

A number of studies such as [2] have found that there is a large turnover among workers in Crowdsourcing platforms. Often, the barriers for turnover is much less steeper than in a traditional employment setting. This often causes workers to either completely stop working in a platform such as AMT or have a partial turnover where they just stop working on tasks from a particular requestor. It is important to address both types of turnover. The former could be illustrative of dissatisfaction with the platform whereas the latter could be due to dissatisfaction with requestors.

We collaborated with CrowdWorks, one of the largest crowdsourcing platforms in Japan. It provides support for more than 200 types of tasks. It is used by more than 25000 companies and close to a million active workers. We begin by analyzing the turnover rate of workers in CrowdWorks. Figure 1 shows what fraction of workers persist with the platform over a period of time. One can see that as much as 75% of workers drop off within couple of months. The number of workers who persist for more than 2 years is as little as 5%. The

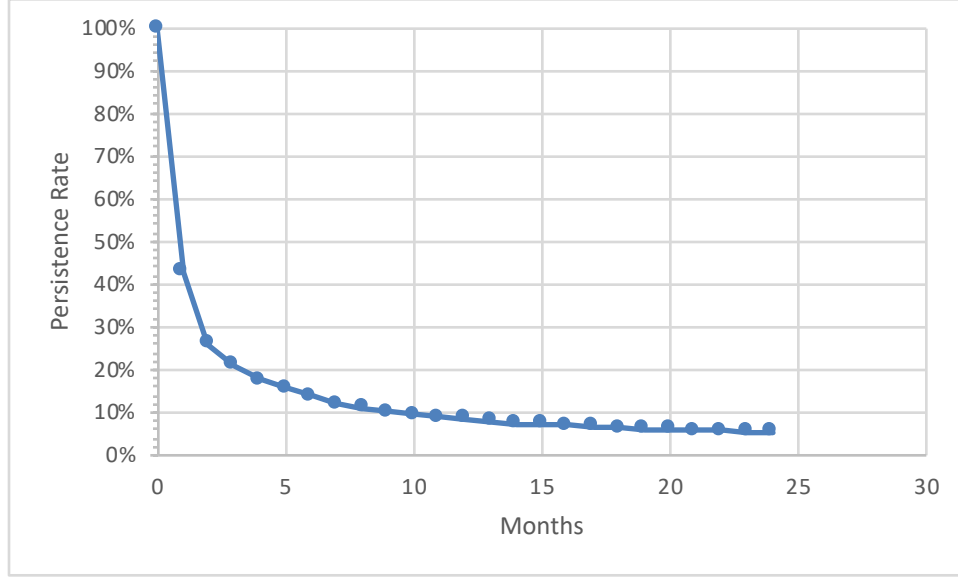


Figure 1: Persistence Rate of Workers in CrowdWorks, according to the log of all workers who made accounts from January 1st, 2017 to December 31th, 2018.

Q1	Platform	Q2					Total
		Very Easy	Easy	Neither	Difficult	Very Difficult	
Yes	AMT	14%	47%	12%	0%	0%	73%
	CrowdWorks	0%	2%	6%	2%	0%	10%
No	AMT	4%	6%	6%	10%	2%	26%
	CrowdWorks	6%	18%	48%	16%	2%	90%

Figure 2: Results of Questions 1 and 2. The result suggests a correlation between the worker mobility and the easiness of switching among crowdsourcing platforms.

success of any crowdsourcing platform hinges on successfully retaining workers. Hence, identifying the factors causing worker turnover and mitigating them is of paramount interest. It is expected that workers who drop-off have a wide variety of reasons for doing so. We are specifically interested in the role of platform design for this phenomenon. By understanding how workers feel about current platforms and their desiderata for new functionalities, one can perform a better platform design.

In December 2019, we conducted a poll of crowd workers on AMT and CrowdWorks, that have different characteristics; AMT is a microtask-based crowdsourcing platform, and CrowdWorks provides a wide variety of task types, focusing on a bit larger tasks such as writing articles and codes. This involved 101 workers (51 AMT and 50 CrowdWorks workers). Figure 2 summarizes the answers to the following questions: (Q1) Do you envision switching between platforms in the next X months? (Q2) How easy/challenging is switching between the platforms? The result clearly shows that there is a strong correlation between worker mobility and the easiness of switching between platforms. We assume that since AMT is a pure microtask platform and obtaining a good reputation is easier, more workers think switching is easy in AMT than in CrowdWorks. Figure 3 shows that workers would like platforms to have many advanced features that are not necessarily fully supported by the current generation of platforms. It is interesting to see that many workers on CrowdWorks dislike the collaboration feature among workers. We have not pursued the reasons yet, but a possible cause is the difference in the granularity of tasks.

Q3. Preference for the feature?	Platform	Answers		
		Like	Dislike	I don't know
Displaying Credentials	AMT	90%	4%	6%
	CrowdWorks	62%	20%	18%
Specifying/ Quantifying/ Learning Skills	AMT	86%	6%	8%
	CrowdWorks	68%	18%	14%
Anonymity	AMT	75%	18%	10%
	CrowdWorks	66%	18%	16%
Complex Workflows	AMT	67%	20%	14%
	CrowdWorks	42%	12%	46%
Collaboration	AMT	84%	4%	12%
	CrowdWorks	28%	42%	30%

Figure 3: Whether workers would like platforms to support for the features or not

3 Platform Design

Based on the analysis of the poll of crowd workers and extensive discussions with participants of the 2019 Shonan Seminar on Future of Work¹, we have identified platform design as one of the key drivers for ensuring the continued success of online job platforms. We first identify a number of issues with the current platform design and make a number of concrete suggestions.

Lack of Interoperability: The current generation of online job platforms operates in silos. Even platforms that are in a specific domain such as driving (Uber, Lyft, Ola, Didi, etc.) are not interoperable with each other. A driver who has driven more than 10K rides with 4.9 rating in Uber will start as a newbie when moving to a different platform such as Lyft. This is also a problem for requesters. Consider a task that requires 10 experts. It is possible that the labor market has the requisite experts – but they are scattered across multiple platforms. In this case, the requester could not successfully complete the task. By enabling interoperability between platforms such issues and many more could be ameliorated.

Lack of Support for Complex Tasks and Workflows: Currently, there are a limited number of tasks for which crowdsourcing is possible. Even sophisticated platforms such as CrowdWorks only support as little as 200 types of tasks. As more and more task types are being serviced by gig economy, the need for supporting more complex tasks becomes important. These complex tasks often have very different set of requirements. For example, they might require a sophisticated workflow so that output of one stage is passed to another. They could require workers with different types of roles. There might also be a need for specialized requirements such as splitting a complex task into multiple sub-tasks that could then be assigned using a workflow.

Limited Pricing Model: There are four types of pricing models that are widely prevalent in online job platforms. First is the fixed income model where each worker is given a fixed amount of money every month or so. There is also task based pricing where the worker is paid a pre-agreed amount after completing a task. There are some intermediate approaches such as fixed income plus a bonus amount and discounted pricing for completing large number of tasks. Finally, there are competitions pricing models in which we pay to the winner. For online job platforms to thrive, it is important to have a wider variety of pricing models.

For the remainder of the section, we propose a number of improvements to platforms that could either reduce the pain points of the key stakeholders or improve their satisfaction.

¹ACM Sigmod Blog, Sept. 23, 2019, <http://wp.sigmod.org/?p=2931>

3.1 Platform Design for Workers

Workers are the back bone of a job platform by completing the tasks of requesters. As discussed in the previous section, the current generation of platforms are not very conducive for long term employment.

Ease of On- and Off-boarding: One of the positive things about online job platforms is the flexibility they provide to the worker. The worker can take tasks that are of interest to them while operating flexible hours. Many platforms simplify them with as little as some identity document and bank account. While joining the platform is straightforward, the onboarding process often leaves much to be desired. Once the worker joins the platform, they are not provided enough guidance to contribute productively. The worker is expected to learn how to contribute on their own. Similar to offline employment in a traditional organization, it is important to have a proper onboarding procedure. The current process for off-boarding is also ad-hoc. Typically, workers and requesters can easily stop working in a platform. However, the workers often lose the reputation that they have earned when moving to a different platform. Similarly, the requesters also lose access to valuable and productive employees when moving to a different platform. It is important to have a better off-boarding platforms so that the workers could transfer the reputation and knowledge learned from one platform to another.

Support for Learning Skills. When a worker joins a platform, she often learns “on-the-job”. If the worker does not perform well due to inexperience, the job could be rejected by the requester thereby affecting the approval rate of the worker. Since a number of requesters filter workers based on task approval rate, this could limit the number of tasks a new worker could contribute to. This often leads to frustration of new workers and eventual turnover. It is often desirable to have a more formal mechanism for simplifying this process. For example, job platforms could have a collection of previously completed tasks that could serve as an on-ramp for the workers. By working on such tasks, the worker can learn the requisite skill in a low stress environment.

Knowledge Base (KB) for Workers: Currently, there are a number of knowledge repositories in an enterprise so that employees know the practices of the company. It is important that online job platforms provide something similar. Note that this is in addition to the aforementioned set of completed tasks. As workers finish tasks, they must be able to add things to a personalized knowledge base about what they learned from the task. This could be public so that any worker can learn from it or private where it is visible only to the worker. As workers become increasingly knowledgeable, this serves as a repository for what they learned over the years. Of course, this must also be interoperable and associated with the worker and transferable as needed. This will also allow workers to find other mentors or experts to learn from.

Support for Expressing Workers’ Preference on Task Assignment The task assignment is usually done by workers themselves, partly because automatic assignment of tasks to workers is not straightforward. There are many reasons for the worker to do the tasks; they did the task because it was easy to do, the task was interesting, they wanted to learn something from the task, it gave them a lot of money, or it was the regular time slot for the worker to do tasks. If the platform has the support for them to express their preferences, platforms will be able to automatically suggest them the tasks more accurately.

3.2 Platform Design for Requesters

In this subsection, we highlight some of the major pain points of requesters and how new functionality from platforms could improve their satisfaction.

Expressive Specification of Task Requirements. Currently, there is limited support from online job platforms for requesters to precisely specify their requirements. For example, in AMT, the requester can filter workers based on approval rate but cannot impose additional sophisticated filtering. It is important for requesters to be able to specify worker requirements such as skills [7], output requirements such as latency, cost and quality. Furthermore, the requester could be open to various tradeoffs such as cost vs quality / latency. For example, one might be willing to pay higher for better quality or faster response. Unfortunately, current platforms do not provide such flexibility.

Supporting Complex Tasks and Workflows. Almost all of the current job platforms support simple microtasks. As more and more tasks are disrupted by the gig economy, it is important to have platforms that can support more complex tasks. Often, complex tasks have a number of distinct requirements. They are often knowledge intensive and collaborative [11] requiring co-ordination with multiple workers. They often are not monolithic and must be split into multiple sub-tasks with different groups of workers completing each of them. They also often require workers to embrace different roles. Finally, they often have a complex workflow where the output of one stage is passed to the next.

Support for Workflow Evolution and Changes during the Execution. Workflows sometimes need to change or evolve during the execution, since completing all tasks requires a long time in general and it is often the case that we find better workflows after we start to execute the original workflow. The platforms should support this kind of evolution and changes of workflows with a minimum amount of effort.

Sophisticated Algorithms for Assigning Workers to Tasks. Currently, most crowdsourcing platforms do not have any sophisticated algorithms in matching workers to tasks. Often, this is done manually by the workers by browsing the list of available tasks. Other than filtering the pool of workers, requesters do not have much control on which workers perform the tasks. Despite extensive research in algorithms for task assignment [1, 4, 11], they are not often incorporated into the platforms. It is important to either have sophisticated algorithms for task assignment so that requester specifications are satisfied or provide an alternate way for the requesters to select workers.

Support for AI Workers. Given the increasing capabilities of AI, it is a matter of time when AI workers become a major part of online job platforms. There are a number of scenarios where AI workers could be useful. If there are urgent tasks, then it is not always possible to use humans to answer them. Often, there is a substantial latency when humans are involved. In these circumstances, AI workers could be a valuable resource. Alternatively, the requester could be cash-strapped and willing to accept less accurate results in exchange for cheaper payments. There are many collaborative situations where AI takes care of the boring work while the human works on the subset that requires human intuition. Finally, AI and humans should be able to replace each other in some situations: an AI can be used as a fallback if a human is answering too late to an urgent task, and conversely, humans can be used as a fallback if an AI fails at recognizing something critical. However, one must be careful in how AI workers are integrated. If not, they could replace human workers causing significant social strife. Furthermore, it is important for requesters to understand the advantages and limitations of the AI workers.

Total Optimization. Currently, most of the work on matching is done in a piecemeal manner where best workers are identified for each task. It is often important to have a holistic optimization that takes the preferences of all the stakeholders into account. This would ensure that good workers are overloaded with work and the workers/tasks are matched in a fair manner.

Algorithm Boutiques: A typical crowdsourcing platform could be improved by incorporating algorithms into the major components including (i) how the task requirements are specified (ii) how tasks are assigned to workers (iii) how the ground truth of tasks are obtained by aggregating worker responses and (iv) how the skills of workers are learned based on their response to tasks. Unfortunately, most of the platforms do not incorporate any of these algorithms. Most of these points are offloaded to the workers and requesters. While experienced requesters often have a concrete mechanisms to effectively achieve each of them, the vast majority of requesters have an ad-hoc and sub-optimal procedure. Finally, even if some platforms implement the algorithms, they are often opaque and the requesters have no say in how they are chosen. It is important for a crowdsourcing platform to have a boutique of algorithms from which requesters can choose the specific variants.

Bespoke Platforms: Most of the crowdsourcing platforms are not very customizable. For example, a domain scientist might need a custom crowdsourcing platform for the task at hand. Currently, the scientist is left with two unappealing choices. Either use an existing platform by approximating the task to suit the constraints of the platform. Alternatively, the scientist could build a new platform from scratch at tremendous cost. In order to unleash the future of work, it is important to enable any requester to create custom online job platforms. It must

have a set of default algorithms that could be customized by the requester. The emergence of on-demand computing frameworks such as Amazon AWS or Microsoft Azure lowered the cost of startups by relieving them of the pressure of managing servers. We believe that the time is ripe to do something analogous for crowdsourcing.

Library of Workflows and AI Workers. In order to help new requesters, it is important for crowdsourcing platforms to provide a large collection of commonly used workflows. Similarly, they could also provide some AI workers that could perform limited set of tasks albeit with the understanding that the work could be of lower quality than that of a human.

Open Source Academic Platforms. Most of the online job platforms are closed source and proprietary. This inhibits research on improving various components of the platform and evaluate the potential impact. One natural solution for driving further research in platform design is through open source academic platforms. A well designed and modular platform could allow a researcher to modify certain components, investigate how it impacts the stakeholders and use to improve platform design. Currently, the researcher has to more or less implement the end-to-end crowdsourcing platform which could be prohibitively challenging. There are a number of promising options such as Headwork and Crowd4U. Headwork² is a proof-of-concept crowdsourcing platform focusing on skill modeling and complex task workflows, using Tuple Artifacts. Crowd4U [9] is a nonprofit open microvolunteering and crowdsourcing platform for academic and public purposes. It has been widely used for tasks such as identifying building damages during natural disasters, annotative tweets, translation, identifying paths of tornados and so on. The most appealing property is its ability to extend the functionality through a datalog type language called CyLog [10]. For example, when the authors came up with an algorithm [11] for task assignment in collaborative crowdsourcing setting they were able to easily implement it on top of Crowd4U [5]. Such functionality has the potential to dramatically improve crowdsourcing research on platform design.

4 Platform Interoperability

In the previous section, we discussed the various functionalities that could be added to the platform to improve it. In this section, we discuss how additional benefits could be obtained by enabling interoperability between platforms. A vast ecosystem has emerged around online job platforms. We begin by creating a taxonomy of such platforms and discuss how these functionalities could be improved and integrated.

4.1 Taxonomy of Online Job Platforms and Ecosystems

So far, the prior work has extensively discussed online job platforms such as AMT or CrowdWorks. However, there is a thriving ecosystem built around these platforms. They often add features missing in the original platforms or provide value added services. In order to improve platform design, it is important to understand this ecosystem.

Job Platforms. For the sake of completeness, we include online job platforms in the taxonomy. These are the platforms such as AMT, CrowdWorks and others where requesters post tasks and workers complete them *online*. These could be generic platforms where a wide variety of tasks are performed or specific ones such as ride-hailing that perform a single task.

Platforms for Worker Communication. Most of the current platforms do not support communications between workers. Some times, this is desirable to avoid inter-worker collusion. Most of the time, this is very limiting as collaborative tasks become more and more important. Furthermore, there are also many scenarios where workers might want to communicate. These include providing feedback about requesters, providing tips and so on. So there is an emerging set of platforms to enable such communication. The most popular of these are Turkopticon [6] and TurkerNation (which closed in 2018). Turkopticon has almost 500K reviews of around 50K requesters. It also provides granular way to provide requestor feedback on fairness, pay quality and speed of

²<http://headwork.gforge.inria.fr>

Support for . . .	Functionalities	Platforms for . . .					
		Providing Jobs	Worker Communication	Worker/Requester Profiles	Task-Worker Matching	Workflow management	Worker Training
Worker	Ease of On- and off-boarding	○		○			
	Learning Skills		○				○
	KB for workers					○	
	Worker Preference Specification	○		○			
	Ease of choosing tasks	○	○	○	○		
Requester	Task Requirement Specification	○					
	Complex Workflow					○	
	Sophisticated Assignment Algorithm				○	○	
	AI Worker Support			○	○	○	
	Total Optimization					○	○
	Algorithm Butiques				○	○	
	Total Optimization					○	○
	Bespoke Platforms	○		○			○
	Workflow and AI Worker Library			○		○	

Figure 4: Relationship between functionalities and platforms for FoW. This suggests that cooperation between different platforms will be the key to make use of the full potential of ecosystem of FoW platforms.

payment. While these are a good start, both Turkopticon and TurkerNation were specific to AMT. It is important that most of these platforms provide a mechanism for worker communication. This could be done within each platform or by providing a generic website where workers from different platforms can congregate.

Platforms for Persistent Worker/Requester Profiles. Currently, each online job platform has an internal mechanism for measuring worker/requestor reputation. This could be as simple as approval rate for workers and requesters. It could also be more granular such as how many tasks a worker has completed for each skill. It is important to have persistent profiles for both workers and requesters so that they can build their reputations based on their entire body of work. Currently, if an experienced worker moves from one platform to another, she has to start from scratch. Hence, it is important to have an independent website where workers/requesters can create a profile and share all relevant information in a verified manner.

Platforms for Matching Workers and requesters. The explosion of online job platforms makes it harder for both workers and requesters. Productive workers and fair requesters are often scattered across platforms. Each worker has to login to multiple platforms to find useful work that quickly becomes tedious. It is desirable to have platforms that could match workers and requesters across multiple platforms. For example, the worker could specify task preferences and the matching platform will scour other online job platforms and notify the worker when relevant tasks become available. This could also benefit requesters by finding productive workers across platforms. While there are some preliminary effort such as by CrowdWorks or Figure Eight that act as a layer on top on AMT, there is still lot of work to be done.

Towards Better and Fairer Crowdsourcing Platforms. Recently, there has been increasing interest in improving crowdsourcing. Some of these efforts include providing guidelines for requesters for specific platforms. Popular examples include Dynamo Guidelines [12] for academic requesters and FairCrowdWork.org. These often help well-meaning requesters to treat workers properly by providing clear instructions about the

task and fair pay. FairCrowdWork.org provides advice to workers based on their rights. They have extensive survey involving 95 questions that allows workers to rate various crowdsourcing platforms. Specifically, they quizzed workers on 8 topics including demographics, work experience in a platform, payment details, communication with requesters and platforms, reliability of the platform, quality of available tasks and miscellaneous information. Based on the response of the workers, each platform is graded in a scale between 1-5.

4.2 Platform Interoperability

Given the plethora of platforms, it is important they are interoperable to ensure that workers and requesters have a wide variety of options. Currently, interoperability is nearly non-existent. However, we believe that this will become more and more important as an increasing proportion of workers join these online job platforms.

There are a number of dimensions in which interoperability must be achieved. Workers should be able to move between platforms without losing any of their skills/ratings etc. This will require the creation of a unified schema of skills. This could be different for each domain. Once this is done, each platform has to fill the worker skills according to this schema. Of course, each platform could have their own additional proprietary list of metric. Nevertheless, it must be possible to create a worker “profile” that gives a holistic perspective of the worker skills in that domain regardless of how many platforms the worker is involved with. Interoperability must also be possible from a requestor perspective. Given a task, it must be possible to recruit workers from multiple platforms. For example, if a requestor needs 10 qualified experts, it must be possible to recruit 4 from Platform 1, 5 from Platform 2 and 1 from a third one. Similarly, it is important to create a requester “profile” that is persistent across multiple platforms so that the worker has a holistic perspective of the requester such as the rate of rejection of tasks.

We can see that achieving interoperability is quite challenging. This requires a commonly agreed standard of skills, which, due to the richness and specialties of the many disciplines that exist, is probably difficult to achieve. For platforms to exchange information, one must design a common, unifying API that will fit for all types of applications.

Standardized API and Schema: The diversity of the crowdsourcing and FoW platforms makes standardization much challenging. However, it is possible that specific components such as task creation, worker selection, truth inference, worker skill estimation could be more amenable for standardization. The creation of the standard API would boost the adoption of crowdsourcing by bringing in more enterprises. Entrepreneurs could also create innovating applications for specific crowdsourcing tasks. This is akin to the explosion of smartphone apps once iOS and Android provided a standardized API functionality. For example, one could identify an important tasks such as Entity Resolution (finding tuples that represent the same real-world entity) and use the API of the platforms to recruit the crowd, perform the tasks and provide the results. There will also be standardization in terms of how metadata about workers and tasks are stored [13, 14].

Interchange Format: There is a need to have a pre-defined interfaces and Schema for data exchanges between platforms. It should provide a simple agreed model that has enough expressive power to achieve total optimization. At the minimum, one must be able to transfer the human factors of workers (such as skill, past tasks, etc) and that requesters (task approval rate, etc). This allows both these types of users to migrate to a different platform without being locked in.

Interchange Queries: When data is not large and is allowed to be exported, platforms can exchange the whole data in the interchange format, but there are cases where this is not a practical solution. An effective approach is to develop common sets of interchange queries so that we can identify only small subsets of data to be exchanged.

5 Conclusion

Online job platforms are becoming increasingly popular. They provide an alternative job arrangement that has the potential to dramatically affect the Future of Work. An increasing proportion of human workforce will be

employed in such platforms. Hence, it is very important to study the problem of platform design. In this paper, we have a preliminary attempt at investigating this problem. We provide a taxonomy of platforms and what are the major missing functionalities for both workers and requesters. We also advocate the need for interoperability between platforms. This allows workers and requesters to move their data between platforms without getting locked into a single platform. There are a number of interesting research and policy challenges in achieving the vision of platform interoperability.

References

- [1] S. Basu Roy, I. Lykourantzou, S. Thirumuruganathan, S. Amer-Yahia, and G. Das. Task assignment optimization in knowledge-intensive crowdsourcing. *The VLDB JournalThe International Journal on Very Large Data Bases*, 24(4):467–491, 2015.
- [2] A. M. Brawley and C. L. Pury. Work experiences on mturk: Job satisfaction, turnover, and information sharing. *Computers in Human Behavior*, 54:531–546, 2016.
- [3] A. Doan, R. Ramakrishnan, and A. Y. Halevy. Crowdsourcing systems on the world-wide web. *Communications of the ACM*, 54(4):86–96, 2011.
- [4] C.-J. Ho and J. W. Vaughan. Online task assignment in crowdsourcing markets. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.
- [5] K. Ikeda, A. Morishima, H. Rahman, S. B. Roy, S. Thirumuruganathan, S. Amer-Yahia, and G. Das. Collaborative crowdsourcing with crowd4u. *Proceedings of the VLDB Endowment*, 9(13):1497–1500, 2016.
- [6] L. C. Irani and M. Silberman. Turkopticon: Interrupting worker invisibility in amazon mechanical turk. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 611–620. ACM, 2013.
- [7] P. Mavridis, D. Gross-Amblard, and Z. Miklós. Using hierarchical skills for optimized task assignment in knowledge-intensive crowdsourcing. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 843–853, 2016.
- [8] S. McFeely and R. Pendell. What workplace leaders can learn from the real gig economy. gallup, 2018.
- [9] A. Morishima, S. Amer-Yahia, and S. B. Roy. Crowd4u: An initiative for constructing an open academic crowdsourcing network. In *Second AAAI conference on human computation and crowdsourcing*, 2014.
- [10] A. Morishima, N. Shinagawa, T. Mitsuishi, H. Aoki, and S. Fukusumi. Cylog/crowd4u: A declarative platform for complex data-centric crowdsourcing. *Proceedings of the VLDB Endowment*, 5(12):1918–1921, 2012.
- [11] H. Rahman, S. B. Roy, S. Thirumuruganathan, S. Amer-Yahia, and G. Das. Task assignment optimization in collaborative crowdsourcing. In *2015 IEEE International Conference on Data Mining*, pages 949–954. IEEE, 2015.
- [12] N. Salehi, L. C. Irani, M. S. Bernstein, A. Alkhatib, E. Ogbe, K. Milland, et al. We are dynamo: Overcoming stalling and friction in collective action for crowd workers. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pages 1621–1630. ACM, 2015.
- [13] D. Schall. *Service-Oriented Crowdsourcing - Architecture, Protocols and Algorithms*. Springer Briefs in Computer Science. Springer, 2012.
- [14] D. Schall, B. Satzger, and H. Psaiar. Crowdsourcing tasks to social networks in bpel4people. *World Wide Web*, 17(1):1–32, 2014.