Collaborative Crowdsourcing with Crowd4U

Kosetsu Ikeda†, Atsuyuki Morishima†, Habibur Rahman‡, Senjuti Basu Roy†⋆,
Saravanan Thirumuruganathan‡‡ Sihem Amer-Yahia◦, Gautam Das‡

†University of Tsukuba. ‡UT Arlington. †⋆NJIT. ◦CNRS LIG. ‡‡QCRI, HBKU.

{lumely,mori}@slis.tsukuba.ac.jp, senjutib@njit.edu, habibur.rahman@mavs.uta.edu,
sthirumuruganathan@qf.org.qa, sihem.amer-yahia@imag.fr, gdas@uta.edu

ABSTRACT

Collaborative crowdsourcing is an emerging paradigm where a set of workers, often with diverse and complementary skills, form groups and work together to complete complex tasks. While crowdsourcing has been used successfully in many applications, collaboration is essential for achieving a high quality outcome for a number of emerging applications such as text translation, citizen journalism and surveillance tasks. However, no crowdsourcing platform today enables the end-to-end deployment of collaborative tasks. We demonstrate Crowd4U, a volunteer-based system that enables the deployment of diverse crowdsourcing tasks with complex data-flows, in a declarative manner. In addition to treating workers and tasks as rich entities, Crowd4U also provides an easy-to-use form-based task UI. Crowd4U implements worker-to-task assignment algorithms that are appropriate for each kind of task. Once workers are assigned to tasks, appropriate worker collaboration schemes are enforced in order to enable effective result coordination.

1. INTRODUCTION

We propose to demonstrate Crowd4U, a prototype system for the deployment of collaborative tasks. Unlike other crowdsourcing frameworks, collaboration is a central tenet of Crowd4U and permeates all its features. Crowd4U provides support for end-to-end deployment of collaborative tasks and enables task decomposition, worker-to-task assignment, and effective worker collaboration during task completion. Requesters can specify collaborative tasks in a declarative manner. Tasks are then assigned to groups of workers taking human factors [11, 9] and worker-to-worker affinity into account. Crowd4U can use any task decomposition algorithm and handles various worker collaboration schemes that ensure effective result coordination. We demonstrate 3 scenarios that represent diverse collaborative crowdsourcing: text translation, citizen journalism, and surveillance tasks.

Enabling Collaborative Crowdsourcing: The ability to support collaborative tasks requires a rigorous formaliza-
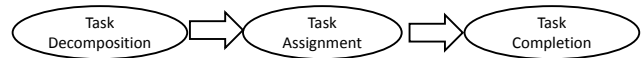
Figure 1: Deployment process for complex collaborative tasks. Result coordination is achieved via worker collaboration schemes in task completion.

tion of the synergistic effect caused by worker collaboration and of other human factors affecting collaboration effectiveness and outcome quality. Worker collaboration takes one of three forms: *sequential* as in text translation [9] where workers improve each others' contributions; *simultaneous*, as in citizen journalism, where workers generate content in parallel; or *hybrid*, as in surveillance tasks, where some workers contribute to fact collection in a sequence, correcting each others' observations, and others provide testimonials separately and simultaneously. Figure 1 summarizes the process of deploying complex tasks. The first step is task decomposition where an input task is broken into smaller micro-tasks. The second step is task assignment whereby worker teams are asked to complete micro-tasks. In the third step, each team gets access to a micro-task and complete it following a task-dependent collaboration scheme. In existing work on task decomposition, requesters are in charge of *result coordination*, i.e., of combining the outcome of different micro-tasks. For example, the Find-Fix-Verify design pattern is commonly used for crowd-powered authoring systems [1].

Crowd4U can use any task decomposition algorithm to break a complex task into micro-tasks. Its key innovations are the reliance of a task-specific assignment algorithm to find the best group of workers to complete a task, and the implementation of *sequential, simultaneous and hybrid worker collaboration schemes*, to ensure effective result coordination. Depending on the nature of the task, the assignment algorithm combines different human factors such as workers' skills, worker-to-worker affinity and upper critical mass [9]. Skills are used to filter out unqualified workers. Worker-to-worker affinity corresponds to the "comfort-level" of workers who are part of a group solving a collaborative task. Upper critical mass is a constraint on the group size beyond which the collaboration effectiveness diminishes. Once tasks are assigned to workers, Crowd4U monitors their collaboration to ensure effective result coordination. The choice of appropriate collaboration scheme is task-dependent.

Comparison with Other Frameworks: While existing research has investigated task assignment for crowdsourcing

in diverse contexts, it often ignored the aspect of collaboration among workers, that is central to the success of complex tasks. For example, well-known crowdsourcing systems such as Deco [8], CrowdDB [3] focus on enabling important primitives in database query processing through micro-tasks that are performed by individual workers whose responses are then aggregated. Other frameworks such as PyBossa [2] or Hive [4] are more generic and can be used for a variety of tasks but are still bound by fixed workflows and micro-tasks with no distinct notion of collaboration. Frameworks such as Argonaut [5] enables complex context-heavy data processing tasks (dubbed "macrotasks") but are still performed by individual workers and not by teams. In contrast, Crowd4U is *declarative, generic* and *collaboration-aware.*

Demo Plan: To illustrate the range of applications that can be enabled by Crowd4U, we organize our demonstration based on the three collaboration types specified above. The application of video subtitle generation and translation highlights sequential collaboration where workers improve the contribution of each other. Support for simultaneous collaboration is exemplified by a citizen journalism application where workers generate report on a specific topic by working in parallel. Hybrid collaboration is showcased by surveillance tasks where some workers contribute to fact collection in a sequence, and correcting each others' observations, while others provide confirmations separately and simultaneously.

Paper Organization: We first describe how the architecture of Crowd4U was revisited to enable collaboration in §2.1. We then describe how task assignment is enabled under different worker coordination schemes in §2.2. The demonstration scenarios are outlined in §2.5.

2. CROWD4U FOR COLLABORATION

Crowd4U[1] is a non-profit all-academic open crowdsourcing platform on which more than 600,000 tasks have been performed for a variety of real-world projects in domains such as libraries, natural disasters, digital archives, cognitive science, and health informatics. Crowd4U was initially designed to support micro-tasks wherein a requester posts a task that is achieved by one worker at a time [6]. In this section, we describe how the platform has been revisited to support collaboration. We first describe the extended architecture, then we explain how tasks are assigned to workers, and finally, show how workers complete tasks.

2.1 Collaboration Architecture

Figure 2 shows the major components of Crowd4U. A requester who wants to register tasks into Crowd4U writes a *project description* in CyLog, a Datalog-like language designed for crowdsourcing applications with complex data flows [7]. CyLog allows humans to evaluate predicates in rules. The rules describing tasks and their dependency are interpreted and executed by the CyLog processor, which dynamically generates and registers tasks into the task pool. Crowd4U also provides tools to help requesters generate CyLog rules by allowing them to define tasks with a form-based user interface and spreadsheets.

Crowd4U can use any task decomposition algorithm to break a complex task into micro-tasks. Its key innovations are the reliance of a task-specific assignment algorithm to
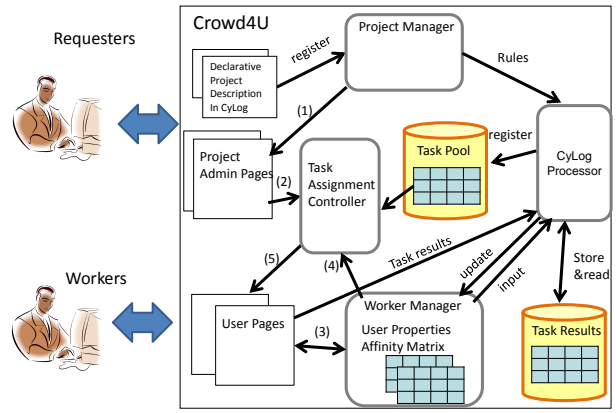
---

[1]crowd4u.org

Figure 2: Crowd4U architecture and workflow for collaborative task assignment

find the best group of workers to complete a task, and the implementation of sequential, simultaneous and hybrid worker collaboration, to ensure effective result coordination. Our task assignment algorithm is scalable [9] and can handle multiple tasks and users. We describe how task assignment and result coordination are achieved.

2.2 Task assignment

A feature of Crowd4U is that the task assignment is conducted in a declarative manner. To make that possible, Crowd4U manages three types of relationships between workers and tasks explicitly. (1) *Eligible* means that a worker is eligible for performing a task. This is computed by the CyLog processor using the project description and worker human factors. For example, in a project description a task requester may specify that only workers who log in to Crowd4U and speak English as a native language are eligible for their tasks. (2) *InterestedIn* means that a worker is interested in performing a task. This is declared by each worker when she is shown a list of eligible tasks. (3) *Undertakes* means that a worker confirms that she performs a task. A (worker,task) pair can go into this relationship status only when the worker is *Eligible* for that task.

2.2.1 Task assignment workflow

Task assignment for collaborative tasks is performed as follows (Figure 2): (1) For each submitted project description, an administration page for the project is generated. The page has a form to enter desired human factors for collaborative task assignment. (2) The entered factors are sent to the *task assignment controller*. (3) User pages show workers the list of collaborative tasks for which they are eligible, and ask them to specify their interest in tasks. The *InterestedIn* relationship is recorded in the worker human factors table maintained by the worker manager. (4) The worker manager supplies the task assignment controller with desired human factors and a worker affinity matrix. (5) The assignment controller chooses a team of workers that satisfies the desired human factors, out of the workers who are eligible and interested in the task. Then, the controller outputs the suggested team and each worker in the team is asked to join the collaborative task.

The assignment algorithm combines different human factor such as workers' skills, worker-to-worker affinity and upper critical mass [9], to find the right team of workers to complete a task. The worker manager supplies the CyLog processor and the task assignment manager with worker human factors (e.g., languages, countries, and application-specific human factors) and the relationship among workers encoded in the *worker affinity matrix*, which maintains the information on how a pair of workers is expected to work well. For example, in the case of surveillance tasks, if workers live in the same geographic area, their affinity value is larger. The assignment controller waits for a sufficient number of workers to show interest in the task before building a team satisfying the requester's desired human factors. Then, the suggested team members are asked to perform the task. Unless all suggested workers start to perform the collaborative task (i.e., to go into the *undertakes* status) by the specified deadline, task assignment is re-executed to find a new team. In addition, if none of the possible teams satisfying human factors accepts the task, Crowd4U suggests to the requester to update her input. Once workers undertake a task, Crowd4U monitors their collaboration for ensuring successful task completion.

Task Assignment Algorithm: Task assignment depends on the nature of the task. We adapt the task assignment algorithm that was proposed in [9] for each kind of collaboration scheme. Intuitively, we model the set of workers as a complete graph with nodes representing workers and edges are labeled with pairwise affinities. A group of workers is a clique in the graph whose size does not surpass the critical mass imposed by a task. A clique with high intra-affinity implies that its members collaborate well with one another. Our task assignment problem reduces to finding a clique that maximizes intra-affinity and satisfies quality and cost limits. [9] proved that task assignment optimization is a NP- Complete problem and hence finding an optimal assignment of worker groups is often infeasible for a large real-time crowdsourcing platform. The authors proposed a series of algorithms that are efficient in practice and provide high quality groups of workers. We adapt the algorithms depending on the type of collaboration scheme. For example, for sequential tasks, we create a single group that maximizes group affinity that also satisfies the quality and cost constraints of the task. For parallel tasks that can naturally be decomposed, we decompose it into a set of independent sub-tasks (such as, independent sections of a document to draft together). We then identify groups for each sub-tasks who edit simultaneously on their allocated section, with collaboration across the sub-groups is needed to effectively merge the sections and prepare the overall document.

2.3 Result coordination

Result coordination between workers is ensured via the following worker collaboration schemes:

Sequential Collaboration: In this mode, the team members collaborate with each other through the tasks dynamically generated based on other members' task results. For example, after a worker translates a sentence into another language, a task for checking the result is dynamically generated, and the result is sent to another team members.

Simultaneous Collaboration: In this mode, Crowd4U first assigns the task to solicit her SNS ID (e.g., Google ac-

count) to communicate with other members in the team. After all the members are in the "undertakes" status, the collaborative task is generated and assigned to all the members with the list of obtained IDs. The members work together with any collaboration tool (e.g., Google docs). The result of the collaborative task is submitted by one of the team members, but recorded as the result produced by the team.

Figure 3: Constraint entry form in an project administration page

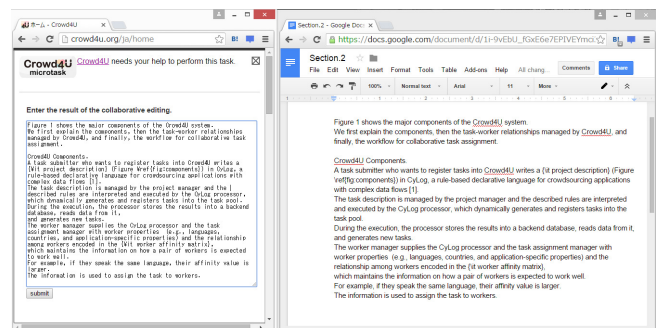Figure 4: Worker human factors on a worker page

Figure 5: Conducting simultaneous collaboration task

Hybrid Collaboration: Crowd4U allows to interleave the two result coordination schemes in a complex data flow. For example, surveillance and correction tasks are executed as a sequential collaboration while the testimonials are provided simultaneously.

2.4 Interface Design and Worker Interaction

Figure 3 is part of screen-shot of the project administration page where a requester specifies the desired human factors for task assignment. The requester also specifies an expiration time for worker recruitment.

Figure 4 shows the set of human factors that can be updated by each worker. Those factors are either provided by the worker when creating an Crowd4U account (e.g., native languages, location) or computed by the system based on previously performed tasks (e.g., via qualification tests, or by learning workers' profiles as in [10]).

Figure 5 is a worker's screen in a simultaneous collaboration where she communicates with other workers using Google doc and submits the result for a Crowd4U task. Crowd4U could be combined with any collaboration tool. While delegating communication methods to other collaboration tools, Crowd4U controls task generation and assignment; it manages relationships between workers and tasks, builds teams, generates new tasks based on the intermediate task results, and assigns tasks to workers.

2.5 Demonstration Scenarios

We plan to demonstrate Crowd4U through of 3 scenarios corresponding to the 3 collaboration schemes.

In the first application, video subtitle generation and translation [9], workers are instructed to first transcribe speech into text in order to generate subtitles in the original language. Then, other workers are asked to translate the resulting subtitles into the target language. It has been shown that for text translation, sequential coordination whereby workers improve each others' contributions, is the most effective scheme [9].

The second application is a citizen journalism one where workers are instructed to write a short report on a topic of their choice (chosen from a list of available topics). Here, workers can work simultaneously, contributing to different parts of the same text.

Our third application is a surveillance task that requires a hybrid coordination scheme. The goal of this task is to collect as much data about facts and testimonials in different geographic regions and at different time periods. Under this scheme, some workers contribute to fact collection in a sequence, correcting each others' observations, and others provide testimonials separately and simultaneously.

We also have a tentative plan to invite demo attendees to wear a requester or a worker hat. As a requester, an attendee will experience how easy it is to create and deploy collaborative tasks in Crowd4U. In order to make that process easy, we will prepare templates that can be modified to suit a requester's need. As a worker, an attendee will be asked to enter her feedback and impressions on the VLDB conference. The resulting document will be made available to other workers to update with their own conference experience.

3. CONCLUSION

We demonstrate Crowd4U, a declarative and collaboration-aware crowdsourcing framework. We also show that Crowd4U can elegantly handle different worker collaboration schemes and thereby deploy tasks in diverse domains, such as, sentence translation, citizen journalism, and surveillance. Crowd4U's declarative and extensible architecture can easily be leveraged to incorporate additional worker collaboration schemes and other task assignment algorithms.

4. ACKNOWLEDGEMENTS

5. REFERENCES

[1] M. S. Bernstein, G. Little, R. C. Miller, B. Hartmann, M. S. Ackerman, D. R. Karger, D. Crowell, and K. Panovich. Soylent: a word processor with a crowd inside. *CACM*, 2015.

[2] R. M. Borromeo and M. Toyama. Automatic vs. crowdsourced sentiment analysis. In *IDEA*, 2015.

[3] A. Feng, M. Franklin, D. Kossmann, T. Kraska, S. R. Madden, S. Ramesh, A. Wang, and R. Xin. Crowddb: Query processing with the vldb crowd. 2011.

[4] N. Gibson and J. Talburt. Hive: crowdsourcing education data. *Journal of Computing Sciences in Colleges*, 25(5):72–78, 2010.

[5] D. Haas, J. Ansel, L. Gu, and A. Marcus. Argonaut: macrotask crowdsourcing for complex data processing. 2015.

[6] A. Morishima, S. Amer-Yahia, and S. B. Roy. Crowd4u: An initiative for constructing an open academic crowdsourcing network. In *HCOMP*, 2014.

[7] A. Morishima, S. Fukusumi, and H. Kitagawa. Cylog/game aspect: An approach to separation of concerns in crowdsourced data management. *Information Systems*, 2016.

[8] H. Park, H. Garcia-Molina, R. Pang, N. Polyzotis, A. Parameswaran, and J. Widom. Deco: A system for declarative crowdsourcing. *VLDB*, 2012.

[9] H. Rahman, S. B. Roy, S. Thirumuruganathan, S. Amer-Yahia, and G. Das. Task assignment optimization in collaborative crowdsourcing. In *ICDM*, 2015.

[10] H. Rahman, S. Thirumuruganathan, S. B. Roy, S. Amer-Yahia, and G. Das. Worker skill estimation in team-based tasks. *PVLDB*, 2015.

[11] S. B. Roy, I. Lykourentzou, S. Thirumuruganathan, S. Amer-Yahia, and G. Das. Task assignment optimization in knowledge-intensive crowdsourcing. *VLDB J.*, 2015.