Crowdsourcing for Enterprises

Maja Vuković
IBM T.J.Watson Research
maja@us.ibm.com

Abstract

Crowdsourcing is emerging as the new on-line distributed problem solving and production model in which networked people collaborate to complete a task. Enterprises are increasingly employing crowdsourcing to access scalable workforce on-line. In parallel, cloud computing has emerged as a new paradigm for delivering computational services, which seamlessly interweave physical and digital worlds through a common infrastructure.

This paper presents a sample crowdsourcing scenario in software development domain to derive the requirements for delivering a general-purpose crowdsourcing service in the Cloud. It proposes taxonomy for categorization of crowdsourcing platforms, and evaluates a number of existing systems against the set of identified features. Finally, the paper outlines a research agenda for enhancing crowdsourcing capabilities, with focus on virtual team building and task-based service provisioning, whose lack has been a barrier to the realization of a peer-production model that engages providers from around the world.

1. Introduction

As the physical and digital worlds are becoming universally connected, and computational resources and data are available beyond their immediate owner, it is now possible to effortlessly reach out to the masses, and open the "*function once performed by employees and outsourcing it to an undefined … network of people in the form of an open call"*, the process which Howe [1] defines as crowdsourcing.

Leveraging the wisdom of crowds in the form of open calls is not a new paradigm. In the past companies have run competitions to engage end-users to contribute towards certain enterprise functions, such as advertising campaign design and specific problem resolution challenges [2]. Knowledge harvesting in the context of support services found in Xerox's Eureka system [3] is an example of an internal crowdsourcing. ReferralWeb [4] allows for content co-creation, while enabling end-users to become aware of their existing communities, which are generated by data mining of Web documents and forums.

Web 2.0 offers a more efficient distributed on-line problem solving and production model, promising benefits of scalable, global workforce, while lowering the cost of execution. A stellar example is the one of The Goldcorp Challenge [5], which successfully employed knowledge of globally distributed geological experts to identify locations of the gold deposit. Numerous purpose-built crowdsourcing solutions are becoming available, allowing enterprises to externalize various stages of their product lifecycle, for example TopCoder.com is enabling crowdsourced software development.

1.1 Objective

At present, there is no general-purpose crowdsourcing platform, which embeds the task-based services to support crowdsourced activites. In order to build such Cloud-enabled service, we require a good understanding of the sheer number of emerging crowdsourcing solutions. In addition we need a structured approach to analyze the capabilities of existing crowdsourcing systems.

1.2 Approach

As a first step towards answering these questions, Section 2 introduces sample usage scenario describing key roles and operations in the crowdsourcing process. Section 3 proposes taxonomy for classifying crowdsourcing platforms and shows how existing crowdsourcing landscape maps into it. It also identifies the set of features for a general-purpose crowdsourcing platform that is used to assess existing crowdsourcing systems. Finally, Section 4 outlines a research agenda for the true realization of crowdsourcing on a global scale.

2. Motivating Scenario

To illustrate how crowdsourcing process can be supported by a general-purpose platform, this section introduces the following scenario in the context of crowdsourced software development.

A small-medium enterprise, ACME, subscribes to crowdsourcing platform CrowdX, which hosts marketplaces and runs competitions. ACME would like to run a competition for a new location-aware resource

978-0-7695-3708-5/09 $25.00 © 2009 IEEE
DOI 10.1109/SERVICES-I.2009.56

discovery algorithm, allowing for example, identification of the server's location at a point in the network that minimizes the total round-trip time among the server and a set of requestors. Along with the description of the task, ACME specifies the duration of the competition, success and evaluation criteria for identifying winning submissions. ACME further attaches a data set for testing. ACME would also like to be able to run the algorithm on a subset of own internal resources, and as part of the task description, exposes the interfaces that would allow providers access to ACMEs testing environment. ACME further specifies their terms, as to how intellectual property arising from this development will be handled. It identifies an award for the winning submission. ACME defines the criteria for filtering out providers to the competition upfront; they have strict certification requirements and minimum level of rating as requirements for participation.

Upon the receipt of the request, CrowdX sets up the competition, establishes interfaces to testing environment, and makes test data available on its file-system. As CrowdX initiates the competition at specified time, they also directly advertise this by proactively matching potential providers with this request.

During the competition time, providers with access to CrowdX access the requirements; socialize them with the ACME and other members of the crowds, through collaboration tools, such as forums and instant messaging.

At the end of the competition ACME assigned judges evaluate submissions. CrowdX executes the payment.

2.1. Roles and operations

Figure 1 shows an overview of roles and their operations in the crowdsourcing process, distilled from the running scenario.

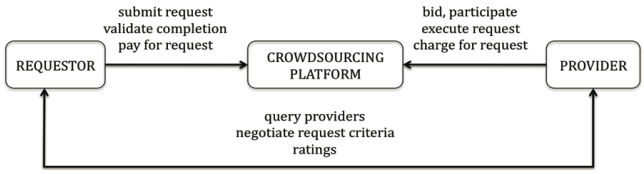

Figure 1 Key roles and operations in crowdsourcing process

Crowdsourcing Requestor is an entity that submits a task request – initiating the process of crowdsourcing, by specifying the acceptance criteria. Requestor pays and awards the successful completion of the task by the service providers. Requestor needs to be able to locate suitable providers after describing their crowdsourcing request using numerous parameters, such as location, skill-set, certification level, rating and reputation, etc. Requestors expect to receive feedback regarding the progress of task completion and potential changes to associated costs (e.g. arising from platform usage). Requestors may also perform management operations on the crowdsourced requests in progress, such as cancelling a particular task or selecting a different service provider.

Members of the crowd undertake the execution of the crowdsourced tasks, often in exchange for monetary rewards. They may use some of the capabilities of the crowdsourcing platform, such as storage, problem determination scripts, access to integrated development environments (IDEs), etc., for which utilization they may be charged.

Crowdsourcing Platform is a trusted broker ensuring that providers successfully complete the task requests and that requestors pay for the charges. Crowdsourcing Platform issues authentication credentials for requestors and providers when they join the platform, stores details about skill-set, history of completed requests, handles charging and payments, and manages platform misuse. Crowdsourcing platform can execute crowdsourcing requests in a number of different modes, by advertising them on the marketplace, allowing providers to bid for them, or in the form of a competition, where requestor identifies criteria to be used for selection of the winning submission. Crowdsourcing platform may further allow requestors and providers to team-up.

3. Crowdsourcing platforms

3.1 Categorization of crowdsourcing platforms

This section presents categorization of existing crowdsourcing platforms, in order to identify the required capabilities of a general-purpose crowdsourcing service. I propose two dimensions: *(1) crowdsourced function* and *(2) crowdsourcing mode.* Table 1 shows a subset of existing crowdsourcing solutions according to the proposed categorization. Not all the platform categorization is discrete.

Crowdsourced function represents the part of the product and/or service lifecycle that is being crowdsourced. Crowdsourced function may take one of the following forms: design, development and testing, marketing and sales and support.

By crowdsourcing design processes, enterprises benefit from innovation and creativity of crowds. Furthermore, they can employ members of the crowd to evaluate their own design, before proceeding with new product development. One of the examples in this category is Threadless.com, a popular community-oriented T-shirt company, where crowds submit their T-shirt designs. Similarly, Japanese design shop Muji used crowdsourcing to both obtain design ideas and evaluation feedback from over half a million people. 99designs is a

platform for designers to complete requested artwork.

	CROWDSOURCING FUNCTION			
CROWDSOURCING MODE	**DESIGN AND INNOVATION**	**DEVELOPMENT AND TESTING**	**MARKETING AND SALES**	**SUPPORT**
COMPETITION	Muji Threadless 99Designs	PeoplePerHour	Marketocracy	Askville by Amazon
MARKETPLACE	Innocentive IStockPhoto	TopCoder Crowdspirit Mob4Hire uTest	PeerToPatent Spot.us Predictify	Amazon Mechanical Turk* GetSatisfaction Fixya

Table 1 Classification of existing crowdsourcing examples by function and mode

iStockphoto is a digital photo marketplace. Innocentive is a marketplace for solving business, science, and product development problems.

The next group of systems supports the product development and testing and reaps the benefits of scalable workforce, and expertise matching. uTest.com, mob4hire.com, TopCoder.com, CrowdSpirit and PeoplePerHour are success stories and example systems in this group.

uTest is the marketplace for software testing services, offering real-world QA services through their community of more than 14,000 professional testers from 151 countries around the globe. Mob4hire, is a company based on the idea which was crowdsourced on the CambrianHouse by Paul Poutanen. Mob4Hire Inc reaches out to 86 countries, 2000 handsets and more than 130 mobile operators, helping mobile application developers access variety of testing platforms and testers in real field conditions. TopCoder.com is a community of over 140,000 skilled software engineers. PeoplePerHour is another platform for software development crowdsourcing. Crowdspirit platform enables businesses to involve innovators from outside the company directly in the design of innovative products and services.

Numerous platforms are used for crowdsourcing of marketing and sales functions. By employing crowdsourcing for marketing and sales functions, enterprises can benefit from crowd analytics. Predictify is a prediction platform that allows news readers to make a prediction on the topic, and have a discussion with other users on-line. Users earn scores for every prediction against the actual outcome, and as a result build a reputation over time based on the accuracy of their predictions. Peer-to-Patent is an effort to open up the patent examination process to public participation and involve community reviewers to assess the patent applications. Marketocracy platform tracks and identifies the best investors and evaluates their trading activity. Spot.us is an open source project, which allows public to commission journalists to do investigations on important stories.

Finally, Amazon Mechanical Turk, FixYa.com, GetSatisfaction.com, and Askville by Amazon are platforms for crowdsourcing supporting functions. They represent examples of community-information centric systems, a next generation of on-line help system that have evolved from traditional user groups and online forums, by integrating social networking capabilities. Amazon Mechanical Turk is a special case as it focuses on providing a marketplace for the micro-tasks, which may include content-creation, testing, and micro development. FixYa.com's community reaches out to (certified) product specialists who provide free technical support and technical help for a range of electronic equipment and consumer products. Aside from posting the solutions to the questions, there is also a collaboration capability built-in, as well as acess to the vendor specific manuals and troubleshooting guides for over half a million products. GetSatisfacton.com is a real-time, dynamic customer support content based on the wisdom of crowds. By crowdsourcing post-sales support, enterprises build collaborative intelligence and increase their own support marketshare. Amazon's askville is a social community site, which facilitates question answering as a game, where users earn or receive reputation in particular topics as they answer questions in those topics, depending on how good their

answer is.

Crowdsourcing mode identifies whether the request is a tender, where providers bid to complete the task, or it is the competition, where a winning submission is selected. Examples include previously mentioned competitions run by Muji for the best design, or similarly by Threadless.com.

Furthermore, crowdsourcing processes can be differentiated based on the way requestors provide incentives, profit sharing and intellectual property rights to providers. In addition, an individual or a team of providers may complete crowdsourced requests. Similarly, the task that is being crowdsourced, may be a microtask, or a complex task, which triggers cascaded crowdsourcing.

3.2. Crowdsourcing process

Global, Web-based, public computing platforms are becoming universally widespread [6][7]. To understand better and expose the challenges and requirements for realizing a crowdsourcing task through a general-purpose, Cloud-enabled, crowdsourcing service the first step is to define the crowdsourcing process. I identify the following four stages in delivering crowdsourcing request, as part of the cloud-computing platform, shown in Figure 2 building on the requirements arising from the scenario described in Section 2.

Registration and specification The process starts with providers and requestors registering with the platform (Step 1, Figure 2). Their identity is verified, credentials issued, and in the case of providers skil1s are evaluated and certified (Step 2, Figure 2). The requestor specifies the task to be crowdsourced, defining its requirements including description of the task (e.g. develop a new algorithm for location aware resource discovery), crowdsourcing mode (e.g. competition), start and end date, and other parameters.

Initialize crowdsourcing contest The platform advertises crowdsourcing request either in the form of a bidding item in the marketplace or an open competition call. During the duration of contest (Step 3, Figure 2), requestors and participants may collaborate and discuss the requirements and approach (Step 4, Figure 2). Participants may team-up to jointly contribute to the request.

Carry out crowdsourcing request Before providers undertake the execution of the crowdsourcing request, the platform acts as a broker between requestors and providers in establishing IP governance (Step 5, Figure 2), legal issues and payment. Once contractual items are agreed upon between requestors and providers, platform sets up the environment and tools to support providers to undertake the crowdsourcing request. This includes provisioning of services required by the task request, if necessary instantiation of the work item and team management tools, etc. It starts by registering and uploading data stores required for the competition (Step 6, Figure 2), followed by allocation of resources required (Step 7) and provisioning of infrastructure for use by crowd (Steps 8). In addition, business support services, such as billing and metering are instantiated (Step 9). For more complex tasks, workflow management may be triggered (Step 10). Finally, licenses for applications used by crowd, e.g. specific IDEs are obtained (Step 11).

Complete crowdsourcing request Once the participant completes the request, submits it to the platform, requestor validates it against the completion criteria. In some cases, e.g. software, algorithms, this process may be automated by employing model checking [10]. While in the case of design requests, this may still involve humans who evaluate and select winning entries.

Once the requestor confirms successful completion of a request, platform executes award payment (Step 13), and requires requestor and providers to rate their crowdsourcing experience. Following the crowdsourcing request completion, any members of the platform, if event is not private, may review and rate contribution (Step 14).

Often submitted contributions may be of low quality or even undelivered (in open marketplaces), which represents a challenge for a successful crowdsourcing process. Furthermore, competitions may not receive any entries, and marketplace tasks may not be bid for. One of the challenges is how to provide suitable incentives (payment) for community to contribute.

3.3 Required features for crowdsourcing

3.3.1 General design requirements

To enable the development of extensible crowdsourcing services, the crowdsourcing platform must be general-purpose, and must not embed any scenario specific dependencies. Furthermore, the system design principles of crowdsourcing must hold independently of the particular infrastructure implementing the framework. To fulfill this requirement the design must allow for extensibility in terms of component technologies, independency of expertise matching algorithms, collaboration services, task management tools, etc.

Crowdsourcing request has a central role in this approach and implies a number of design requirements. Firstly, the framework must allow a requestor to select

the complexity of the task and to parameterize its description in terms of the business and technology requirements. Another important challenge in providing a crowdsourcing is dealing with failures, e.g. because of the loss of network connectivity.
When deployed the crowdsourcing framework will be exposed to a large number of concurrent crowdsourcing requests and providers. The design of the framework must ensure its ability to operate under increasing load, increasing complexity of requests and increasing size of participant community.

Furthermore, the heterogeneity of networked connected services and resources used by the large variety of requestors and providers providing request data increases the risk of security compromises. Finally, crowdsourcing platforms gather and distribute personal information about individuals. It is essential that users have means for retaining control over the distribution and dissemination of their private information.

3.3.2 Crowdsourcing Features

Regardless of the crowdsourcing request type, mode and domain of application there are some basic, common and shared process steps in building crowdsourcing service to support it, addressing the four stages of the crowdsourcing process.

Registration and specification

1. Registration facility allowing service providers to join the platform on-line and in a matter of seconds, both in the role of requestor and provider (including authorization and issuance of credentials).
2. Comprehensive model of the crowdsourcing request, including the description, expected outcome, duration, expiration, quality parameters, incentives, etc.
3. Mechanism for defining and supporting both micro- (e.g. image tagging) and complex tasks (e.g. development of a mobile application).
4. Identity federation and profile porting, allowing requestors and providers to automatically port their on-line profiles, skill-sets, reputations from other systems.
5. Request templates allow a) novice requestors to start a task in a matter of seconds, and b) expert requestors to create multiple parallel identical requests.
6. Crowdsourcing platforms gather, collate and distribute personal information about individuals. It is essential that users have means for retaining control over the distribution and dissemination of their private information.
7. A mechanism for defining flexible resource pricing schemes for resources, which are required for crowdsourcing request completion.

Initialize crowdsourcing request

8. Ability to support multiple crowdsourcing modes (e.g. run competitions as well as act as a trusted broker in the marketplace).
9. Collaboration services (e.g. forums, instant messaging, etc.) allowing for requestors and providers to socialize crowdsourcing request requirements and negotiate terms in real time.
10. Support contract negotiation through integration with IP governance services.
11. Real-time, proactive discovery of the providers, based on the their skill-set, ratings, etc.
12. Virtual team formation that employs expertise matching mechanism and discovered social relationships.
13. Dynamic knowledge integration platform that provides efficient search, captures the user experience, and supports collaborative knowledge evolution.
14. If dictated by the crowdsourced request, capability to integrate with the business process (internal infrastructure of the enterprise).

Carry out crowdsourcing request

15. A single, easy-to-use interface allowing a requestor to monitor and manage multiple providers and request progress across the platform.
16. Fine grained accounting for participation and consumption of resources.
17. A mechanism for auditing of providers, generating a complete historical view of crowdsourcing process.
18. Interfaces for mobile devices, as required by specific tasks (e.g emergency responses) for crowd to contribute from the field.
19. Technology provisioning: to allow for anyone in the crowd to participate, provide tools and services to access for the task.

Complete crowdsourcing request

20. Ratings and review service, capturing the performance of providers and requestors.
21. Automated submission evaluation to facilitate efficient evaluation of completed crowdsourcing requests.
22. Automated provisioning of crowdsourced request deliverable, if applicable. For example, completion of crowdsourced software component may trigger service registration process.

3.4 Assessment of existing platforms

Table 2 Assesmment of existing crowdsourcing systems

	Crowdsourcing system			
Capability	**MTurk**	**Innocentive**	**iStockphoto**	**Threadless**
Mode	Marketplace	Marketplace	Marketplace	Competition
Function	Design, Support	Design, Innovation	Design	Design
1	+	+	+	+
2	+	+	+	+
3	-	+/-	-	NA
4	-	-	-	-
5	+/-	+/-	+/-	NA
6	+	+	+	+
7	NA	NA	NA	NA
8	-	-	-	-
9	+/-	+	+	+
10	-	+/-	+/-	+/-
11	+/-	+/-	NA	NA
12	-	-	-	-
13	-	-	-	-
14	-	-	NA	NA
15	+/-	+/-	NA	NA
16	-	-	NA	NA
17	+/-	+/-	NA	NA
18	-	-	NA	NA
19	-	-	-	-
20	-	-	-	-
21	+	+	+	+
22	-	-	-	-

Legend: + feature exists, +/- partially supported, - feature does not exist, NA = not applicable to use case

Table 2 shows the extent to which four different crowdsourcing platforms, namely Amazon's Mechanical Turk, Threadless, Innocentive, and iStockPhoto, meet the specific subset of technical requirements for delivering an end-to-end general-purpose crowdsourcing capability, as outlined in Section 3.3.

Amazon Mechanical Turk [8] is a crowdsourcing marketplace where providers undertake microtasks, such as image tagging and content production, in an exchange for a monetary prize.

Innocentive is a crowdsourcing platform, where providers received cash awards for solving challenging research and development scientific.

Threadless is an example of an on-line t-shirt store, which runs competitions for t-shirt design. Submitted designs are put to a public vote. The selected designs contribute towards new printed t-shirts, and designers receive cash awards and store credit.

iStockPhoto is a digital image marketplace (now expanding to various multimedia forms). Contributors are pre-screened, and need to pass certain quality level.

These four systems have been selected as they are representative each of different crowdsourcing function and mode.

All of these platforms are purpose-built, and support a specific crowdsourcing function, except for InnoCentive, which accommodates for a range of design and development functions. Most of the existing crowdsourcing frameworks fall short of facilitating dynamic formation of globally distributed teams. First step towards such a capability can be found in PeoplePerHour, which enables end users to create teams from their social network, and then to jointly bid on projects. However, a more flexible and proactive team discovery and building based on a multiple criteria of

past, current and future contexts (e.g. completed requests, skill set, and ratings) is lacking.
Furthermore, none of these crowdsourcing platforms provide comprehensive set of tools and computational services that can be used by crowds to participate in problem resolution and production. The inaccessibility to technology and services required for crowdsourcing tasks, represents a major barrier to engaging crowds from all geographies.

4. Research Agenda

The actual realization of the promising benefits for enterprises from crowdsourcing are far from being well-achieved and pose an extensive range of interesting challenges along social, legal, economical and technical dimensions as Brabham [9] identified. From the technical perspective, building crowdsourcing platforms opens up several systems design and interaction challenges: opening up data to communities, building and maintaining communities of interest, enabling collaborative intelligence, facilitating massive multi-tenancy, dynamic discovery and service integration, dynamic knowledge integration and economic models for incentives.

Following is the set of research challenges for the next generation of crowdsourcing systems:

1. Crowdsourcing request model, including not just desired award and completion time for a microtask, but also more complex criteria and the ability to pool a number of providers supplying different aspects of a request.
2. Dynamic and symmetric request and participant matchmaking: how to effectively process and evaluate requestor request and match it against the repository of existing requests, as well as (teams of) providers?
3. Providers should have the tools to evaluate bids based on complex criteria, not just immediate incentive.
4. Dynamic pricing component: software components can be developed, which will adjust the pricing of crowdsourcing requests according to demand and participant availability.
5. Modelling and evaluating the quality of experience in crowdsourcing process.
6. Designing a mechanism for virtual team formation, incorporating not only skill-set, but also discovered social networks.
7. Integration of computational services supporting the request completion in the cloud environment.

5. Summary and Outlook

The existing crowdsourcing systems are often purpose-built, supporting a set of specific, micro tasks in a particular domain and a specific part of the product lifecycle. To address the identified gaps in architectural support for building crowdsourcing service I have embarked on building a platform for crowdsourcing, demonstrating how the Cloud infrastructure can be used as a scalable hosting and application development environment for dynamic, task-based crowd teaming.

6. References

[1] J. Howe. *The Rise of Crowdsourcing*, Wired, 14(6), URL (accessed 23 October 2008): http://www.wired.com/wired/archive/14.06/crowds.

[2] J. Surowiecki. *The Wisdom of Crowds.* Anchor. (2005).

[3] D. G. Bobrow, J. Whalen. *Community Knowledge Sharing in Practice: The Eureka Story*. Journal of the Society of Organizational Learning and MIT Press, Volume 4 Issue 2, Winter 2002

[4] H. Kautz, B. Selman, M. Shah. *Referral Web: Combining Social Networks And Collaborative Filtering.* Communications of ACM 40, 3 (Mar. 1997), 63-65. 1997.

[5] GoldCorp Challenge. URL (accessed 14 November 2008). http://www.goldcorpchallenge.com/

[6] K. Delic, M. Walker. *Emergence Of The Academic Computing Cloud.* Ubiquity Volume 9, Issue 31, August 5 - 11, 2008.

[7] C. Hewitt. *Orgs For Scalable, Robust, Privacy-Friendly Requestor Cloud Computing.* Internet Computing, IEEE. Volume 12, Issue 5, Page(s):96 – 99, Sept.-Oct. 2008

[8] A. Kittur, E. H. Chi, B. Suh. *Crowdsourcing User Studies With Mechanical Turk*. In Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems (Florence, Italy, April 05 - 10, 2008). CHI '08. ACM, New York, NY, 453-456.

[9] D. C. Brabham, *Crowdsourcing As A Model For Problem Solving: An Introduction And Cases*. Convergence: The International Journal of Research into New Media Technologies, Vol. 14, No. 1, 75-90, 2008.

[10] McMillan, K., Symbolic Model Checking, Kluwer, ISBN 0-7923-9380-5.