Challenges and Experiences in Deploying Enterprise Crowdsourcing Service

Maja Vukovic[1], Jim Laredo[1], and Sriram Rajagopal[2]

[1] IBM T.J. Watson Reserach Center, 19 Skyline Dr, Hawthorne, NY 10532, USA
{maja,laredoj}@us.ibm.com
[2] IBM India, Chennai, India
srirraja@in.ibm.com

Abstract. The value of crowdsourcing, arising from an instant access to a scalable expert network on-line, has been demonstrated by many success stories, such as GoldCorp, Netflix, and TopCoder. For enterprises, crowdsourcing promises significant cost-savings, quicker task completion times, and formation of expert communities (both within and outside the enterprise). Many aspects of the vision of enterprise crowdsourcing are under vigorous refinement. The reasons for this lack of progress, beyond the isolated and purpose-specific crowdsourcing efforts, are manifold. In this paper, we present our experience in deploying an enterprise crowdsourcing service in the IT Inventory Management domain. We focus on the technical and sociological challenges of creating enterprise crowdsourcing service that are general-purpose, and that extend beyond mere specific-purpose, run-once prototypes. Such systems are deployed to the extent that they become an integrated part of business processes. Only when such degree of integration is achieved, the enteprises can fully adopt crowdsourcing and reap its benefits. We discuss the challenges in creating and deploying the enterprise crowdsourcing platform, and articulate current technical, governance and sociological issues towards defining a research agenda.

Keywords: Enterprise crowdsourcing, Governance, Crowdsourcing Process.

1 Introduction

With the realization of Web 2.0, the trend of harnessing large crowds of users for mass data collection [1] and problem-solving [2], has become a growth industry employing over 2 million knowledge workers, contributing over half a billion dollars to the digital economy. Crowdsourcing is nowadays being employed in many domains, ranging from advanced pharmaceutical research [3] to T-shirt design [4].

Two types of methods have been employed to catalyze the crowd participation. Firstly, the more traditional ones, rely on motivating the participants to share information and thereby either gain creditability or get reciprocal information [1, 5]. Secondly, many systems provide tangible incentives to crowd participants for their contributions (e.g., monetary prizes) [3,4,5,6].

Tapscott and Williams [7] discuss how businesses can harness collective capability of outside experts to facilitate innovation, growth, and success. In contrast, our research investigates applicability of crowdsourcing methodology within the enterprise, thereby engaging internal networks of knowledge experts.

Section 2 describes the IT Inventory Management problem to which we have applied enterprise crowdsourcing service. Section 3 discusses our experience in developing and deploying an enterprise crowdsourcing service. We present challenges in building a general-purpose service, which integrates with the existing business processes, and the necessary development support to facilitate shorter turn-arounds in customizing new use cases. Finally, we discuss incentives and governance issues. Section 4 outlines a set of challenges for the true realization of enterprise crowdsourcing.

2   Use Case: Crowdsourcing for IT Inventory Management

IT inventory management captures and manages the enterprises' IT assets in numerous repositories, which are often outdated and incomplete. Thus they often fail to provide consolidated, global, views of the physical infrastructure and actionable data (e.g. which business applications would be affected if a specific data center is consolidated). However, this information can be found in the core of the organization - the knowledge workers that understand and drive the business and the IT itself. Yet, there is little transparency on who knows what within an enterprise. Locating such critical business information becomes intractable, especially as the desired knowledge is transferred between experts when they transition within the organization.

To address this IT inventory management challenge within a large enterprise, we used crowdsourcing ("wisdom or crowds") approach to discover, integrate and manage the knowledge about the physical infrastructure that hosts *business applications*. An example of such a business application would be a company's support website, which resides on a number of servers. As such, business application is differentiated from the actual middleware that may support such as a web server or messaging queue, as those can be discovered by utilizing advanced scripts.

The goal of our crowdsourcing service was to find out and verify the information about 4500 business applications. We have idenitified initial enterprise crowd from the existing business application registry. Before sending of the e-mails with task requests, we verified that the targeted crowd was reachable and still in the same role within the enterprise. Where possible, we used 'delegate' application owners. Additional feature of our crowdsourcing service was the capability to re-assign tasks to other team members.This was useful in the scenarios where a) different team members possessed partial knowledge about the task and b) business application owner was no longer responsible for the given application.

 For each business application contributors were asked to:

1) Verify the application ownership
2) Provide compliance information (e.g. is the application subject to ITAR?)
3) Identify servers that are hosting it (e.g. enter fully qualified hostname) and their type (e.g. production, development, testing, etc.).

There were three exceptional scenarios that were also considered: 1) application has already been sunset (it's no longer running on the infrastructure), 2) application is hosted on a 3rd-party server and 3) application is not hosted on any server (e.g. application could be a spreadsheet file on a destkop).

Using the enterprise crowdsourcing service, we harnessed an expert network of 2500 application owners to execute the IT Inventory Management exercise - gathering information on the mapping of 4500 business applications to more than 14,000 of IT Systems (servers). Crowdsourcing service has achieved 30X improvement in process efficiency, in contrast to the traditional approach of employing two full time experts to manually reach out to the application owners and gather the information. This process is rather time consuming, and typically would result in 100-200 applications being captured over the two months time. Furthermore, the powerful knowledge network, generated as a result of crowdsourcing run, can be situationally engaged for other large-scale business and IT transformation initiatives, such as cloud transformation.

3   Experience in Deploying an Enteprise Crowdsourcing Service

Not all the crowdsourcing use cases within the enterprise are the same. Inherently they impose a set of requirements on the actual crowdsourcing process. For example, is the task to be initially sent to one or more experts? Can the task be separated in a number of (concurrent) subtasks? What support for task sequencing needs to be supported? Can the user reassign the task or parts of the task to other users? How do we create tasks, what sort of existing business repositories are available and can serve to initiate the crowdsourcing tasks? Finally, how does one go about validating the collected data? In this section, we discuss the crowdsourcing process, its elements, and key requirements for the development support to enable efficient deployment and customization of new use cases.

3.1   Task Management and Crowdsourcing Process

When designing our crowdsourcing service we have been striving towards an approach that allows drawing best practices and eventually leading us to build a self service system where we can reuse our key elements. The use cases we have identified are of a knowledge seeking nature, such as the IT Inventory Management one that we describe in this work. We build a business process that facilitates the capture of knowledge, usually around a business object (e.g. a business application or a server).

We look at the business process as a sequence of steps where people either contribute knowledge to the business object or make decisions upon the knowledge that has been captured, such as validation, request for more information, or simply complete the step in the process. A step in the process we define as a task.

In its purest form, crowdsourcing tasks should be available for everyone to apply and attempt to complete them. In an enterprise environment we have the benefit of other intrinsic knowledge as part of the organization that allows us to pre-assign tasks to initiate the process. We call this the seeding process. It uses some information

about the business object, such as prior, current ownership, or lead process owner, for example. Once the tasks are assigned, task owners may provide requested knowledge, segment the work and refer segments to different parties, or simply forward to someone else to take care of the task. One concept under consideration is the use of optional fields, if the capture process allows it, we may want to capture information as quick as possible, yet not delay key steps in the process when prior ones have been completed. By using optional fields we can expose and attempt to gather other information, yet if it is not captured by the time the mandatory information is captured, we may proceed to the next step.

Task ownership allows for other support services to accelerate the completion of the tasks, such as the use of reminders and escalations. Once a task has been referred to, there is the possibility of a delay due to lack of attention or misdirection. It is important to raise awareness and force an action as soon as possible. In our studies we were able to complete 50% of the tasks in 4 days [8], and with use of reminders and escalations we were able to keep the process alive and close almost 90% of the tasks in the following 3 weeks.

The lessons learned have helped us improve the task management process, and we are driving to a design that allows us to define any business object and apply the task management on top of it.

As tasks are completed, our system captures who is modifying each field of the business object and completing tasks. This information allows us to build an audit trail of all changes. This information creates user community around the business artifact as a by product of the process. This user community has many dimensions, It could be around a particular business object, or around the skills required to complete the task. The community can be invited or a member can be referred to help on an open task. The community usually outlives the process, and can be invoked at a later stage when rerunning the business case or for new extensions of the process.

Finally, crowdsourcing process requires certain governence support. Various use cases, may call for the admistrative and mediator roles, to provide capabilities such as task cancellations, task management or resolution of conflicts (e.g. esp. on the marketplaces where task requestor offers monetary rewards).

3.2   Implementation of the Enterprise Crowdsourcing Service

We have build our enterprise crowdsourcing service as a Web-application, designed to support knowledge acquisition activity. It enables task creation based on existing business artifacts, such as a task for the existing application in the business application registry (e.g. task for the "Support Website"). Tasks are either assigned to single or multiple users (based on ownership information available in the business artifacts). The task list for each user is displayed in the home page upon logging into the crowdsourcing service. When tasks can be worked upon by multiple users concurrently, an application level lock is introduced to enable concurrency. Users are notified whenever the task they are working on is concurrently modified. This helps multiple users collaborate on a single task in a transparent manner. Users are also presented with the modification history of artifacts enabling them to know who changed information and what changes were made.

Our crowdsourcing service relies on the Enterprise Directory (that is, a list of employeed, their contact details, and organizational structure) for user authentication and for obtaining the user profile information, such as such as first name, last name, location and manager's details. It further utilizes Enterprise Directory to enable quicker and easier lookup of employee names and e-mail IDs when re-assigning tasks. Business artifacts are sourced from existing internal repositories, such as server and business application registries. Enterprise crowdsourcing service relies heavily on the recency of data in such applications as tasks are assigned to users based on the same. Wherever required, it validates such data against the enterprise dricetory to confirm validity.

Enterprise crwodsourcing service also features reminder capability, to send out e-mails to users with tasks pending for more than a predetermined number of days in their task list. The reminder schedule can be customized by the administrator.

The application also enables users refer part of or the entire task to other users. Preferred fields of the task can be selected before sending a referral and these are highlighted when the referred user works on the task. Auto complete of referral emails in the email selection box is enabled for easy and quick user lookup against the enterprise directory.

Administrative users are provided with the capability to manage tasks by enabling them to cancel, complete or reassign open tasks; reopen canceled and completed tasks and editing task information without changing ownership. Administrators can also view reports and logs of the modification history of the artifacts.

3.3   Development Support for Building Crowdsourcing Solutions

As a crowdsourcing application needs to manage tasks in a context different from that of traditional workflow systems – multiple users collaborate on a single task with the ability to forward part of or the whole task to one or more users. Hence, the persistence layer requires a framework or tool that provides transaction isolation and transaction transparency – each user working on a task needs to know what information has been contributed/ modified about the task by other users and who did the same. Transaction transparency can be achieved on the persistence layer by alerting the user carrying out modifications about any intermediate transactions. Since tasks go through various stages of its lifecycle, development would be easier if the persistence layer also supports state-modeled artifacts.

The data layer should provide efficient techniques to cache and retrieve look-up and drop down list data that are used frequently. This is especially critical when the crowdsourcing platform caters to capturing information about physical assets in the organization which typically number hundreds of thousands in big organizations. Also critical is the efficiency of generating reports based on various criteria as the data size can run up considerably. The servers hosting the application should be clustered to achieve load balancing and failover.

The application also requires access to the enterprise's employee look up database to validate if tasks are forwarded to valid users and for ease of lookup of employee details such as email id, name and location. Access to details such as the project of the task owner would be helpful in cases where no response is obtained from the owner

so that other members in the project may be assigned the task. Integration with other organizational systems also facilitates in developing a good crowd sourcing system by providing data on assets, language preferences of users, whether the user is still active within the organization or is on short/long leave during task creation etc.

The application also requires a robust email service which can handle the load required by the system. The email service should provide templating capabilities to send out personalized and customized email messages. Some organizations have restrictions on sending mass mailers to employees within the organization. The support of management is essential as crowdsourcing systems usually involve sending out high numbers of emails to employees – by way of initial information seeking emails, referral emails and reminders for pending tasks.

3.4   Incentives and Governance

The success of any collaborative endeavor, both within an organization or when an outside community is engaged, heavily depends on the incentive mechanisms put in place to achieve the desired outcome. Numerous incentive mechanisms exist and make collaborative production challenging [9]. Traditional award schemas are presently employed by enterprises, e.g. salary, performance bonuses. Incentives for crowdsourcing raise new legal challenges (e.g. compliance and taxation).

Types of Incentives
When building a crowdsourcing service, there are two types of incentives that designers need to consider. Firstly, the challenges is how to attract new members? Secondly, and more importantly, the question is how to encourage the contributions and sustain the community of the time?

Effective integration of new members is critical factor to successfully building an online community. Challenges with attracting and enabling contributions of new members can be grouped in the following categories [10,11,12]:

1. New members tend to be less committed
2. They need to learn the norms of the behavior
3. Need to understand the structure of content and engagement.

To encourage contribution in a crowdsourcing exercise, one needs to reduce any entry barriers, and provide clearly identified goals. A critical factor for participants, aside from the clearly identifiable goal and skillset, is that benefits outweught the cost of participating. Cosely et al. [13], present an intelligent task routing mechanism to increase contribution by targetting contributors'efforts to where they are more needed, and based on their expertise.

Incentives can further be grouped into material and non-material (social) ones. Examples of material incentives, include monetary, such as the ones employed by Mechanical Turk and Innocentive. In the same group, are (material) prizes, which are utilized by TopCoder and Netflix for example, to award the best performing members of the crowd.

Our experience

The deployed crowdsourcing application   included capabilities that allowed participants to collect virtual points for crowdsourcing task, as well as for any successful referrals. The points, however, at this stage were not exchangeable for a tangible, material award. The users had the ability to see their rating, compared to other participants. Finally, many users considered access to this consolidated repository, which was a result of the crowdsourcing, to be an incentive on its own.

When introducing an incentive for a crowdsourcing task within an enterprise, a number of questions are immediatelly raised: how would incentives affect employees of different status (contractor vs. full-time), or similarly how does one engage a service-type employee (with a billable utilization rate), as opposed to the traditional employee with a flat-rate compensation. Following is a set of questions that form a research topic within the enterprise crowdsourcing domain:

1. How are compliance, taxation and labor laws addressed as crowdsourcing is being adopted?
2. How do all of the above apply to a global enterprise that coexists in many national/regional jurisdictions?
3. How do we give incentives in a global company across borders?
4. How does a company view employees doing work (e.g. crowdsourcing tasks) that is not their "day job"?

4   Challenges

Deploying enterprise crowdsourcing service provides an insight into its design and use, but deploying systems beyond the limited existing, specific- prototypes will require significant progress toward integration with the existing business processes. We consider major challenges that must overcome before achieving this goal. This list is by no means exhaustive; it provides a set of open questions that define the requirements on components of enteprise crowdsourcing services.

1. Process: How does crowdsourcing become an extension of the existing business process?
2. Technical: How can we rapidly develop and deploy new use cases for enterprise crowdsourcing?
3. Sociological and governance: What are the effective incentives that can be deployed within the global enterprise, and what implications do they have on the existing compliance, tax and labor laws?

References

1. Olleros, F.X.: Learning to Trust the Crowd: Some Lessons from Wikipedia. In: Proceedings of the 2008 International MCETECH Conference on E-Technologies (2008)
2. Brabham, D.C.: Crowdsourcing As A Model For Problem Solving: An Introduction And Cases. Convergence: The International Journal of Research into New Media Technologies 14(1), 75–90 (2008)

3. `http://www.innocentive.com`
4. `http://www.threadless.com`
5. Bobrow, D.G., Whalen, J.: Community Knowledge Sharing in Practice: The Eureka Story. Journal of the Society of Organizational Learning and MIT Press 4(2) (Winter 2002)
6. Kittur, A., Chi, E.H., Suh, B.: Crowdsourcing user studies with Mechanical Turk. In: CHI '08 (2008)
7. Don, T., Wikinomics, W.A.D.: How Mass Collaboration Changes Everything. Portfolio Hardcover (December 2006)
8. Vukovic, M., Lopez, M., Laredo, J.: People cloud for globally integrated enterprise. In: The 7th ICSOC. 1st International Workshop on SOA, Globalization, People, & Work, SG-PAW 2009 (2009)
9. Bartlett Christopher, A., Ghoshal, S.: Transnational management: text, cases, and readings in cross-border management, 3rd edn. Irwin/McGraw Hill, Boston (2000)
10. Kraut, Burke, Riedl, van Mosh: Dealing with newcomers. Working paper 12/7/07 (2007)
11. Arguello, J., Butler, B.S., Joyce, L., Kraut, R., Ling, K.S., Ros, C.P., et al.: Talk to me: Foundations for successful individual-group interactions in online communities. In: CHI 2006: Proceedings of the ACM Conference on Human-Factors in Computing Systems, pp. 959–968. ACM Press, New York (2006)
12. Bryant, S.L., Forte, A., Bruckman, A.: Becoming Wikipedian:Transformation of a Participation in a Collaborative Online Encyclopedia. In: Proceedings, GROUP '05, Sanibel Island, Florida, November 6-9 (2005)
13. Cosley, D., Frankowski, D., Terveen, L., Riedl, J.: Suggestbot: Using intelligent task routing to help people find work in wikipedia. In: Proceedings of the 12th ACM international conference on intelligent user interfaces, ACM Press, New York (2007)