RC25272 (WAT1203-055) March 23, 2011
Computer Science

IBM Research Report

Collective Intelligence Applications in IT Services Business

Maja Vukovic, Osamuyimen Stewart
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

Collective Intelligence Applications in IT Services Business

Maja Vukovic and Osamuyimen Stewart
IBM T.J. Watson Research
19 Skyline Dr, Hawthorne NY 10532, USA
{maja, ostewart}@us.ibm.com

Abstract—**IT outsourcing enables companies to contract out IT services, such as infrastructure and application management to external providers. IT services delivery relies on knowledge that is in collective possession of application and infrastructure specialists. With recent advances in harnessing the expertise of network-connected humans, services businesses have also started to seek strategies to efficiently utilize the collective knowledge of their employees. In this paper, we present the application of collective intelligence to three different service types: 1) automation (translation solution), 2) infrastructure management (asset inventory discovery) and 3) application management (software development). We extrapolate a set of salient properties (i.e., input, output, and size of the collective input) as the key elements for employing collective intelligence within the services business. We discuss some of the differences amongst the disparate collective intelligence services and present the resulting distinctive properties as challenges to inspire further research in services business.**

Keywords-component; service delivery, automation, data quality social networking

I. INTRODUCTION

From a customer's perspective, IT Services, in the most general sense, refer to a distinguishable, measurable, orderable and chargeable unit of service which provides a required capability.

While IT Services are often associated purely with providing IT management capabilities on behalf of the outsourcing client, they in reality span a wide range of functions, from automation capabilities to business process services.

Table 1 shows loose grouping of a wide range of services based on their function. Examples include order tracking service, storage service, desktop service, Websphere support service, SAP consultancy service, etc.

Table 1 captures one way of cataloging IT services based on the enterprise function that they deliver, such as: automation, infrastructure, application, management and governance.

Type	**Examples**
Business Process	HR, Governance, Compliance,
Services	Consultancy
Application Services	Application Development and Management, Middleware Services, Database Services
Infrastructure Services	Network Services, Service Operations, Storage Services
Automation Services/Solutions	Transaction Processing, Tracking, Translation, etc.

Table 1. IT Service Types

However, their design, delivery, operation and optimization capabilities rely on the enterprises' core strength -- specialists who understand and drive the business and the IT. Indeed, with the proliferation of globally distributed operations, enterprises are seeking new ways of tapping into these powerful expert networks within and outside the enterprise firewall. The nature of the IT service and its dependency on human experts at different stages of service evolution (design to operational optimization) defines the type of knowledge sought after (e.g. atomic data, such as infrastructure parameters or complex models, such as user practices) and how the experts are engaged.

Today's computing systems are increasingly engaging humans for their intelligence and skills as part of challenging tasks. This technique, often referred to as crowdsourcing, is generally described as a Web-based approach for harnessing the collective intelligence of a large network of individuals. As crowdsourcing (along with all of its trappings and extensions) gains wider acceptance in the public domain, individual service providers inside the global enterprise have started to adopt aspects of the strategy by employing collective intelligence to improve internal business processes or for solving specific problems, thus resulting in a class of enterprise-based crowdsourcing applications-- where the participants are (contractually) affiliated with the enterprise, as opposed to being members of the general public.

While several challenges abound in harnessing the collective intelligence inside the enterprise such as the need to conform to human resource regulations (e.g. those affecting the incentive structure) and other business control parameters [9], however, the goal of this paper is to provide a better understanding of how attributes from different organizational designs can be put together to derive an optimal framework for crowdsourcing in IT services across Service Types.

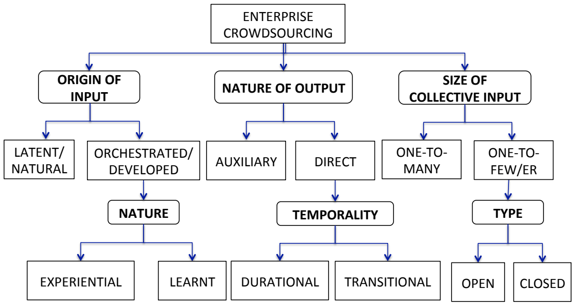

Figure 1 Framework of Organizational Design in Enterprise Crowdsourcing

Figure 1 captures the proposed overall organizational design of enterprise crowdsourcing, denoting the interplay between the collective behavior amongst groups of individual actors and the attendant intelligent behavior of the group in problem-solving or improving business processes. We have coined the term "shades of collective intelligence" to describe this kind of organizational behavior in global enterprises.

When Service Types (Table 1) are plugged into a collective intelligence organizational framework (Figure 1), the function provided by a service and its complexity suggests that there are two parameters that drive the collective intelligence efforts: size of the collective effort (number of experts), and type of knowledge (input) that is to be provided by the experts. Size of the collective effort decreases as we from automated to process-type services, due to increased customization and specialization levels of services often only a handful of experts may posses the sought-after knowledge elements.

On the other hand the type and complexity of the knowledge that can be harvested in different service types increases as we go from automated to process based services. At the automation level, such knowledge elements are often atomic and represent operational characteristics of the service. At the process level, knowledge is more tacit and we often need to consider more open-ended insights about user practices.

The paper makes three key contributions:

1. Analyzes our experiences from the organizational designs involved in deploying three different types of crowdsourcing applications (for three Service Types), and from which the similarities (salient properties) are offered as key elements of the organizational framework for enterprise crowdsourcing.
2. Identifies a set of distinguishing properties of shades of collective intelligence within the enterprise, by distilling origin of skill, input and output to the collective process.
3. Defines a progressive evolutionary pattern (trend) in the organizational design of collective intelligence systems.

The paper is structured as follows. Section 2 puts the paper in the context of ongoing developments in collective intelligence within the enterprise. Section 3 describes our experience from deploying crowdsourcing as part of three different IT service types. Section 4 defines shades of collective intelligence in the context of the enterprise. Section 5 discusses effects of crowdsourcing, motivation, and quality considerations in the context of the three different service types. Section 6 concludes with future research directions.

II. RELATED WORK

Employing collective intelligence to improve internal business processes is not a new paradigm, although technological advances have scaled the potential of its reach. In the past, companies were engaging employees through competitions (e.g. employee of the month) and surveys to get more insights about the potential improvements in the workplace, or to sample ideas for a new product or service. Early examples of collective intelligence are found in systems such as Xerox's Eureka system [1] where human knowledge is harvested to assist with functions in the domain of IT support services.

The ubiquitous access to computing systems has rapidly turned Malone's vision [4] of fluid workforce into a reality. Enterprises are becoming dynamic creatures that seamlessly integrate intelligence and skills of scalable workforce of in-house and external experts, often on-demand.

TopCoder is an example of a crowdsourcing platform specializing in software development competitions. It engages a community of over 250,000 software engineers worldwide, where any number of developers may participate in a single TopCoder competition, where only the best two will win a prize. All submissions are treated the same whether it took the member one hour or ten hours to complete [3]. Clients contract TopCoder, which in turn employs large network of publicly accessible engineers, to develop a specified computer application.

Similar mechanisms are employed by enterprises internally. Tata Consultancy Services [2] built their own platform for enterprise crowdsoucing to improve the efficiency of internal software engineering processes. Analogous to TopCoder, in-house talent, such as new trainees and experts not fully utilizing their technical skills, are being exposed to challenging tasks, introducing the disruptive resource allocation model. The proposed system enables a reputation model, as the means of motivating the participation of in-house experts.

As the digital generation (also known as Gen Y) joins the workforce en masse, enterprises are redefining work environments. Many companies are embedding elements of gaming into their internal crowdsourcing processes. For example, at Microsoft, Smith [7] designed a productivity game to engage employees from around the world to use their extra time and language abilities to assess localized versions of the Windows operating system in different languages. By framing the business goal in the form of a game, it is easier to communicate the objective to the predominant Gen Y employees and increase their participation and contributions. Similarly, in order to perform often mundane task of software testing of new products, specialists are using a game-based mechanism to earn points for their contributions and feedback, which are transformed into a real dollars donated to disaster-relief agencies.

III. ORGANIZATIONAL DESIGNS

In this section we describe our experiences from applying crowdsourcing to three different types of IT services, which harness the collective capabilities of employees for specific work function, ranging from translation in n.Fluent [9], knowledge discovery in BizRay [10] to software development in IT-Stage. We present each system in terms of problems it addresses, and the design (participation requirements) which consists of incentives provided, and outcomes of the deployment.

A. n.Fluent

Service Type: Automation

n.Fluent system is an IT service solution that provides self-service automates translation using statistical modeling translation (SMT) approach. It allows enterprises to quickly translate electronic documents and Web pages—even live, instant messages exchanged on smartphones.

Problem addressed:

nFluent uses crowdsourcing to effectively tap into the collective intelligence of multilingual employees to translate sentences or correct machine translated sentences for improving translation accuracy and quality.

Design:

n.Fuent's basic entry requirement into its translation portal is for employees to be proficient in one or more of its eleven foreign languages and English. Due to dialectal and linguistic issues, proficiency is determined during user registration based on self-identification as a native speaker, without any formal gateway or validation prior to performing the translation task.

Incentives:

To engage language experts within the enterprise, n.Fluent provided a spectrum of incentives, ranging from monetary donations to charity, recognition (reputation) on leader board, assignment of "virtual island" with personalized language-game, and localized-culturally-driven reputation (recognition by Country General Managers)—driven by automated scoring system based on the specified duration of the context/challenge. In addition, the reputation earned within the community is advertised through the complementary award of internal electronic "Thank You" award to enable career development of the winners and participants.

Outcomes:

n.Fluent has successfully engaged and nurtured a vibrant and active world-wide pool of about 8,000 volunteer translators in the eleven languages. Since 2008, more than 36 million words have been translated by the community through various timed "translation challenges" and individual submissions, which are massaged for improving the SMT, engines using BLEU [6].

B. BizRay

Service Type: Infrastructure

IT Management and Optimization processes such as asset management, maturity assessment, or business process transformations, require detailed inventory of IT systems and dependency analysis to name a few. BizRay [11] is an enterprise crowdsourcing service based on principles of 'wisdom of crowd' to accelerate knowledge discovery about IT and business within the enterprise.

Problem addressed:

BizRay, for example, enables rapid generation of a snapshot of the state of IT systems (e.g. servers) and operations (e.g. compliance to US International Traffic Arms Regulations, know as ITAR) at a short notice. Typically, this "non-discoverable knowledge" is gathered in semi-automated way, which at best provides crude estimates, and doesn't scale in large global organizations. Furthermore, with the wide adoption of global delivery model, such knowledge is in the collective possession of globally distributed team members.

Design:

BizRay employs crowdsourcing to quickly design a process solution for a family of business objects, gathering required knowledge stored in the collective possession of in-house specialists. As such, the pool of potential participants in this effort is often defined by their work function. BizRay relies on existing repositories of employees and their work function and/or responsibility for a particular business process. However, even though an organization may be maintaining some list of employees by work functions, it is often out-of-date and/or incomplete, as employees move within and out of the organization. Consequently, crowdsourcing is applied to harvest not only employees' knowledge of a business matter, but also their knowledge about other team members and their competencies. Thus, a by-product of BizRay campaigns is an up to date registry of specialists and their work functions.

Incentives:

In BizRay, members were "competing" on individual basis, despite its collaborative approach to knowledge acquisition. The portal embedded a point-based framework to track participation, both in the form of knowledge contributions (task completion), and also in the form of users' capability to identify a better-suited expert to complete a given task (task reassignment). Whilst the points were not exchangeable for tangible awards, as no tangible awards were offered to participants (due to business/organizational constraints), yet access to accumulated knowledge has been shown to be the key driver for contributions. In addition, when sending reminders for incomplete work items in BizRay we have embedded few times more details about the progress about the overall crowdsourcing campaign, which has further amplified effectiveness of the reminders and resulted in an even higher response rates.

Outcomes:

BizRay has been deployed as part of 10 different business process transformations in Services Delivery and IT Optimization domain, engaging over 10,000 employees, to accelerate knowledge gathering through collaborative and distributed approach. There are two key benefits to BizRay: one is improved quality of data through structured knowledge collection, second is reduced lapse time in knowledge discovery. BizRay has demonstrated ability to improve process performance up to 30x [10].

C. *IT-Stage*

Service Type: Application/Management

IT-Stage provides application management services where in-house experts often supplement staff of outsourcing clients, co-manage or manage and co-develop or develop any type of application-from legacy mainframe systems to Web-based and custom applications, plus off-the-shelf packaged solutions.

Problem addressed:

IT-Stage is an initiative that uses crowdsourcing strategy to accelerate software development work in-house and enhance growth opportunities for software development professionals.

Design:

As part of the IT-Stage program, professionals with 'time in between assignments' or available 'free cycles' can locate and register to complete 'short cycle' work posted to the Portal. "Short cycle work" represents a specification for a work product or deliverable that can be completed within 5 to 10 days. It represents a 'one time' work assignment, and is defined in a manner consistent with the component-model (for software design or development activity) or based on well-partitioned units of work (for whitepapers, requirements documents or other activities). The strategy is to tap 'free cycles' of professionals in the open market through a methodology that yields high quality results on an 'outcomes-based model'.

Incentives:

IT-Stage uses performance specific metrics to reward users' participation, directly impacting employees evaluation. Aside from cash awards, participants are able to use the hours that they spent on participation in the crowdsourcing task towards their utilization rates. Moreover, by developing robust, reusable software components, participants were building their online reputation and demonstrating their abilities, which can lead towards career development. In addition, contributions accumulated by members of one department are used to rank the performance of the different business units. IT-Stage therefore exercised both individual and group level incentives.

Outcomes:

The main result of employing crowdsourcing strategy in IT-Stage was a significant improvement in the utilization rates of software engineers. .

In addition, given its open nature, many developers often chose assignments outside of their expertise to demonstrate their (newly acquired) skills, and use the online reputation built to advance in their career.

IV. SHADES OF COLLECTIVE INTELLIGENCE

We postulate these three disparate deployed applications, described in Section 3, as exemplifying different degrees of collective intelligence derived from the enterprise expert networks. Accordingly, we distinguish (a) origin or nature of input to the crowdsourcing system, (b) its output, and (c) the size of the collective input, as anchoring parameters in the spectrum of collective intelligence within a global enterprise, as shown in Figure 2.

Williams and Ross [11] categorize the work tasks that are amenable to productivity games into core (skills that do not differentiate users), unique (job specific) and expanding (what employees aspire for). Orthogonally they borrow the principles from organizational science, and further categorize work functions by: a) in role (directly impacting employees pay) and b) organizational citizenship behavior (OCB) [12] (what the organizations would like employees to exhibit). They identify that *tasks that expand in-role*

skills and *tasks that stimulate OCB leveraging core skills* are the most favorable for effective productivity games.
In our framework for organizational design in enterprise crowdsourcing, shown in Figure 2, we take a different approach to classifying employees' skill by first observing the origin (or nature) of the input. There are two kinds of (human) input to an enterprise crowdsourcing system: it could either be latent (those that can be acquired through personal natural capabilities of the employee) or orchestrated.

Furthermore, we differentiate two types of orchestrated skills; one is acquired through work experience (e.g. knowledge about the client). By contrast, the other type of an orchestrated skill is acquired through learning, which may not necessarily be done on the job (e.g. how to program in Java).

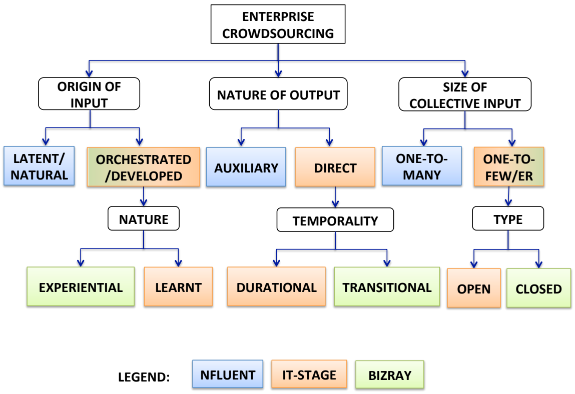

Figure 2 Framework properties of the 3 deployed applications

Next, is the nature of the output of the enterprise crowdsourcing effort which may be auxiliary or complementary i.e., an effort that transforms the human skill into a data element, or it may be direct, i.e., an effort that forms a piece of knowledge base or becomes a component of a tangible product (an end state). We observe the temporality of the direct product of crowdsourcing as consisting of the durational (e.g. piece of software whose coding procedure essentially remains unchanged), and transitional (e.g. knowledge about compliance regulations that apply to client's account, which can easily be forgotten by the expert when they move on to a new role).
Finally, we note the size of the collective input, which describes the cardinality of the relationship between humans (participants) and a crowdsourcing task such as one task being performed by many participants versus one to a few or fewer participants. In addition, the enterprise crowdsourcing tasks that engage smaller communities (few or fewer participants) typically take the form of either an open or a closed campaign. Open campaigns are setup for tasks that require orchestrated-durational skills, while closed campaigns involve tasks that require learned-transitional skills.

We now turn to the discussion of how the different properties of the 3 deployed applications map into this framework for organizational design in enterprise crowdsourcing, and observe the relationships between different elements.

nFluent employs latent skills that are typically acquired through personal background and interest (language learning), and for a single token (a work item or single sentence) may engage up to N employees for translation. The completed token of the translation task is an *auxiliary output,* a product that feeds into the end-state of the crowdsourcing campaign (e.g. a translated word becomes a part of a sentence or a paragraph). From an organizational design standpoint, the cost of setting up such crowdsourcing campaigns is often low due to it's low entry barriers for participation, which triggers extremely high levels of participation, along with the ability for contributors to work at any time. Finally, since the token from the crowdsourcing instance is not an end state, the quality considerations are more relaxed (lower) because the "bad seeds" can be weeded out in the secondary process or from aggregation of the collective intelligence [2].

IT-Stage relies on the software development skills of participants, which are often acquired through job training, but may also be learnt through personal development (e.g. Java programming or Android application development), skills that we term *orchestrated-learnt*. The output of IT-Stage, a software component delivered to the specifications, is the end state of the crowdsourcing campaign, which we term *direct output.* This output is *durational,* meaning that it will not decay over the time, as the requirements for a particular skill does not drastically change over the time (e.g. Java programming will still be the same, although capabilities may be enhanced over the time). Therefore, the token produced from this form of crowdsourcing campaign is relevant over longer periods of time. Since the software development skills can largely be acquired either at school or in personal time (e.g. interest in a new technology) crowdsourcing campaigns harnessing this skill are open (e.g. not too restrictive with respect to the participants' expertise). From an organizational design perspective, these campaigns have higher cost of execution and quality considerations: it relies on semi-experts, which reduce the pool of participants, and the crowdsourcing token (end state instance) produced is also restricted by the one-to-few constraint. The quality considerations are much higher than n.Fluent (latent-based) applications as the eventual end state requires the engagement of (external) experts to review the submitted pieces of code.

BizRay engages a handful of highly specialized experts *(one-to-fewer)* to collect, often confidential knowledge that is derived from participant's exposure to a specific work function (e.g. IT inventory management). The nature of the input requires *orchestrated-experiential* skills, which are highly dependent on the work function of the participant, and cannot be acquired without a business process context. This restricts the nature of the community that can contribute to such campaigns, characterized by what we term as closed. In terms of organizational design, BizRay has the highest cost of execution and quality considerations, given the highly specialized participants whose time is limited and expensive. Contributors to these campaigns produce *direct output* (an end state) but one that is *transitional,* due to the changing nature of the business knowledge and practices. By implication, this transitional nature also increases the cost of execution. For example, while Peter was responsible for an ACME client account they didn't have to conform to US International Trade and Arms Regulations (ITAR) because he "knew" the work-around. However, since Peter transitioned to a new role, changes in the business controls have impacted the regulations for a number of client accounts. Therefore, Peter's particular knowledge/skill and the *direct output* produced are no longer valid; they have decayed over the time.

Figure 3 provides a visualization of the foregoing discussion by capturing the relationship between origin of input (skill) and the desired level of participation to achieve the anticipated output, and putting it in the context of the three different applications we have deployed.

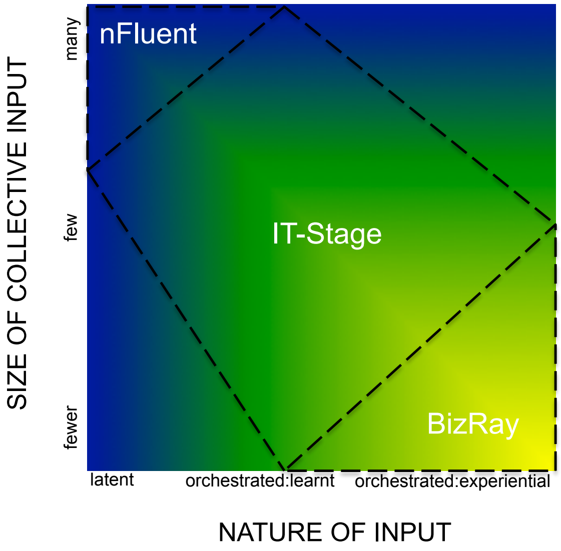

Figure 3 Shades of Collective Intelligence in Enterprise

V. Discussion

In this paper, we have described our practical experiences with three different systems for enterprise crowdsourcing that altogether have harnessed the collective intelligence of over 30000 employees.

We defined the sphere of collective intelligence in the enterprise by three parameters a) input b) output and c) size. As a consequence of this organizational framework, we discussed different quality considerations and associated process for deriving the tokens in the types of crowdsourcing instance including multiple submissions, sequential verification and human evaluation.

In this final section, we will now discuss the consequences and some of the challenges of the organizational framework with respect to the crowdsourcing applications.

A. *Evolutionary trend in the organizational design of collective intelligence*

We have observed a trend in the evolution of organizational design of collective intelligence crowdsourcing systems inside the enterprise: a progression from simple (basic, unitary) systems to more complex (multipurpose) systems based on the view of the composite features (externally) derived from the examination of all three applications.

Whereas successful examples of crowdsourcing such as TopCoder (leveraging learned skills like software development) already existed in the public domain, several similar crowdsourcing attempts in the enterprise were largely unsuccessful.

By contrast, crowdsourcing involving latent (natural) human skills are amongst the first wave of successful systems across several enterprises (IBM, Microsoft, Sun, etc).

The more recent wave of applications has explored new domains or areas with the crowdsourcing strategy such as BizRay (IBM), Productivity Games (Microsoft), etc. We postulate that this logical sequence of successful deployments is not merely an accident of time; rather we take it as indicative of the evolutionary chart of organizational design of enterprise crowdsourcing. We conjecture that this is heavily linked to the nature of incentives central to the organizational design:

- OCB incentives are easier to design and communicate since they are more likely to appeal to a wider audience
- Learned (semi-expert) skills like software development, although fairly common across employees, require greater rigor and commitment in the design, and also less likely to generate a "crowd" compared to OCB-type systems.
- Yet, by contrast, the semi-expert skills-system is a lot easier to design than the more complex system involving expert skills.

Given these observations, it appears that the trend for the organizational design of collective intelligence is to start with the more basic (common) task (which usually draws the crowd), and then evolve to more complex tasks.

B. Motivation: Reaching out to the community

Aside from the work function and type of knowledge that the different crowdsourcing applications gathered, they also applied distinct approaches to reaching out to the crowd. Malone et al. [5] observed that material reward result in fast response, while non-material rewards increase the quality and attentiveness to the task at hand.

nFluent used targeted advertising on the Intranet, with periodic updates, to drive the global translation challenges. This included promotional videos and also localized notifications in specific subsections of internal enterprise Web portal. Besides direct campaign through email and newsletter by the community manager; many n.Fluent participants promoted this campaign through word of mouth capability, provided in the form of an “invite a friend” button on the n.Fluent page (this is collected in user records with intent to use it in overall score). This social networking feature rewards those who invite the most “friends” from their network to n.Fluent. Finally, what was unique to n.Fluent was the ability of certain organizations or divisions to credit the volunteered time of their employees under the condition that they work on the organization specific content, thus making it a quasi-version of IT-Stage

In BizRay, the communication was initiated via e-mail sent to the pre-determined set of users. Potential employees to be targeted were chosen by their work function, and often derived from existing expertise repositories. E-mail was often sent from an executive-level highlighting the importance of the business activity. Regular reminders have resulted in the increase in responses. In addition, escalations (e.g. sending e-mail reminders with the inclusion of the management chain) have been introduced to further increase the responses and raise the importance of the data collection efforts. Each participant can invite other team members or experts to help complete the knowledge requests, by selecting “delegate” feature within the BizRay Web application.

In IT-Stage, communities of software engineers have been engaged through e-mail communication, again management driven, within each department. Similarly to n.Fluent, participants in IT-Stage have the ability to invite and advertise specific software development competitions to their social network.

C. Incentives: Ensuring crowd participation

Participation of large-scale networks is the main influencer of the success of collective intelligence efforts. As outlined in section 3, a wide range of incentive schemes were used in the organization design, often classified into material and non-material, to encourage valuable human input.

The nature and design of these three systems open several questions regarding the incentives: how do we design incentives for tasks that involve a single individual, as opposed to a group? How do we motivate new contributors and sustain existing ones? Finally, how do we cater for different demographics,? For example, younger generations value feedback in the loop and transparency of the work, expecting fairness in the system. Similarly, how can we accommodate the impact of cultural background on the effectiveness of incentives in a global enterprise?

D. Quality Assurance: Getting a bang for the buck

Quality has been the main concern about the effectiveness of the human computation techniques. Questions are raised about potential malicious contributions of incorrect and even potentially harmful results. What is really the level of quality needed to successfully consume the output of enterprise collective intelligence efforts?

In n.Fluent, since the crowdsourcing token (translated words) is not an end-state and also because the quality of translation is not accessed in real time (but done offline using BLEU), the strategy (consistent with the organizational design) was to encourage high volume of contributions. Indeed, such high volume was necessary for crowdsourcing contributions to have any real significant impact on the SMT engine, which relies (requires) on millions of words. The level and quality of contribution varied across the eleven languages (e.g., those with different dialects or experiencing a faster pace of demographic changes saw a corresponding affect on the quality of submitted translations). Given all of these, the level of quality is not uniform across the languages. In some of the more active languages with the most participation, we have seen a range between 5%-20% improvement on the models.

In BizRay, which manages vast number of streams of information pertaining to the core business functions quality becomes even more pronounced. During the deployments we faced several challenges. Firstly, we defined the level of quality by the % of correctness based on number of same responses given by different users answering same question. For example, we asked application owners to list the servers that host their applications. We would then ask the system administrators owning these servers to validate the input. Based on the sample of 200 responses we established 80% of correctness in data. This is, however, still an improvement over the level of quality of data previously manually collected that at best provides crude estimates, by handling responses through e-mail and chat, and even phone conversations and tracking the chain of responses.

Secondly, due to the overwhelming number of requests, e.g. quite often a number of different data collection campaigns may overlap and users would be inclined to ignore subsequent requests. Another concern about the quality of the data, and a challenge for future work is the freshness of the data provided. Whereas in n.Fluent data responses may not age over time, due to the persistence of ground truth, in

BizRay we deal with operational data about dynamic IT environments.

IT-Stage has a very well defined quality measurement drawing on the requirements specifications and derived test cases for software components. In addition, they evaluate the design of the component in terms of its reusability. This however does come at the cost of having dedicated technical experts evaluating the code; and poses the question of the value proposition of crowdsourcing. For example, at what point does this approach no longer deliver the economies of scale, by the mere fact of having to pay dedicated reviewers as the number of submissions rise. More recently top performers from the crowd can take the role of the code reviewer. Can we proactively learn about users and automatically test the submitted contributions. Beyond the competitions, users are also internally rated on the time taken to deliver a specific component.

VI. Conclusion

As we move to the next step of collective intelligence, integrating it with digital intelligence to support large-scale business processes, we identify a number of research questions. As crowdsourcing becomes even more appealing and effective to enterprises the immediate goals include automated discovery, definition and quality assessment of tasks.

Looking towards the future of collective intelligence inside the enterprise, what is the blueprint of a proactive process diagnostics mechanism, which necessarily embeds process driven crowd engagement?

Quality control has so far been investigated under the assumptions that users collectively contribute their knowledge to a singular work item, often isolated from the context of the business process and its overall objective. This opens up a research question of how do we design effective incentives that encourage high quality contributions of one or group of individuals to an orchestrated set of different atomic work items that comprise a business process?

Finally, in this paper we focused on crowdsourcing within the enterprise firewall. With the adoption of global delivery model, complexity of service systems is continuously evolving given the diversity of service providers and consumers. This fundamentally impacts how we get services to work together, and how experts across organizations are engaged. With no single entity owning the end-to-end service system, the challenge is how the collective intelligence is managed in complex service networks.

References

[1] Bobrow, D. G. and Whalen, J. (2002), Community Knowledge Sharing in Practice, The Eureka Story, *Journal of the Society of Organizational Learning and MIT Press*.

[2] Jayakanthan, R. and Sundararajan, D., (2011), Enterprise Crowdsourcing Solution for Software Development in an Outsourcing Organization, *Second International Workshop on Enterprise Crowdsourcing,* International Conference on Web Engineering.

[3] Lakhani, K., Garbin, D., and Lonstein, E., TopCoder (A), (2009), Developing Software through Crowdsourcing. *Harvard Business School Case 610-032.*

[4] Malone T. (2004), The Future of Work. How the New Order of Business Will Shape Your Organization, Your Management Style and Your Life, *Harvard Business School Press*.

[5] Malone, T., Laubacher R., and Dellarocas C., (2009), Harnessing Crowds: Mapping the Genome of Collective Intelligence, *Working Paper No. 2009-001,* MIT Center for Collective Intelligence.

[6] Papineni, K., Roukos, S., Ward, T., and Zhu, W. J., (2002). "BLEU: a method for automatic evaluation of machine translation" in *ACL-2002: 40th Annual meeting of the Association for Computational Linguistics,* pp. 311–318

[7] Smith, R., (2010), Using games to improve productivity in software engineering. *In Proceedings of the ACM SIGKDD Workshop on Human Computation (HCOMP '10).*

[8] Stewart O., Huerta J. M., and Sader, M., (2009), Designing crowdsourcing community for the enterprise. *In Proceedings of the ACM SIGKDD Workshop on Human Computation (HCOMP '09).*

[9] Stewart O., Lubensky D., and Huerta J. M., (2010), Crowdsourcing participation inequality: a SCOUT model for the enterprise domain. *In Proceedings of the ACM SIGKDD Workshop on Human Computation (HCOMP '10).*

[10] Vukovic, M., Laredo, J., and Rajagopal, S., (2010), Challenges and Experiences in Deploying Enterprise Crowdsourcing Service, *International Conference on Web Engineering..*

[11] Williams, J., and Smith, R., (2009), Score One for the Quality! Google Test Automation Conference. Zurich.

[12] Wikipedia. Organizational Citizenship Behavior.http://en.wikipedia.org/wiki/Organizational_citizenship_behavior