Managing Quality of Human-Based eServices

Robert Kern, Christian Zirpins, and Sudhir Agarwal

University of Karlsruhe, Karlsruhe Service Research Institute, Germany
{robert.kern,christian.zirpins,sudhir.agarval}@ksri.uni-karlsruhe.de

Abstract. Modern business strategies consider Web-based outsourcing of microjobs to the masses. Respective business activities are however difficult to manage. Traditional approaches of covering human tasks in business processes build on assumptions of limited process scale and closed organizational models that are not valid in crowdsourcing scenarios. Web services have been proposed as a means to represent human tasks that allow leveraging interaction, brokerage and composition capabilities of SOC for human interaction management. In this paper we argue that crowdsourcing requires considering qualitative constraints and sketch a platform for managing the quality of human-based eServices.

Key words: Crowdsourcing, Human-based eServices, QoS Management

1 Introduction

Despite all endeavours of rationalizing their business processes, organizations still face many types of tasks which cannot be fully automated but which require human intelligence or action to be completed. Traditionally, such activities are integrated into workflows as so called *human tasks*, which are passed to a well defined group of employees of the company. The drawback of this approach is its limited scale. A large workforce has to be hired to cover peak workloads that partly idles in times of small workload resulting in cost overheads.

Amazon was first to address this issue by offering the Web marketplace *Mturk* (`http://www.mturk.com`), on which service requesters can publish open calls for *Human Intelligence Tasks (HITs)* [1]. Any Internet user that meets certain skill criteria may act as a service worker and complete tasks for small amounts of money. The business model of the platform is to act as a broker between requesters and workers and to keep a percentage of the service fee for each task. However, there is no control over the correctness of the results that are delivered by the services workers, nor about any other quality aspects like response time or availability. In case a certain QoS-level is required, the requester has to take care of it himself. This limitation dramatically restricts possible usage scenarios.

In this paper we argue that the utilization of human-based electronic services for large-scale outsourcing of human tasks requires platform support for actively managing their QoS and offer initial considerations on respective platform mechanisms. In the following sections, we will first elaborate on the economical background of human-based electronic services and introducing the term *people*

service for a broad type of human tasks in sec. 2. Following that, we highlight some requirements for respective service-oriented platforms and infrastructure mechanisms in sec. 3. Finally in sec. 4, we outline our conceptual solution approach for an integrated platform that does not only act as a broker between service requester and service worker but which extends the business model by actively managing the QoS of the results delivered to the service requesters.

2 Human-based Electronic Services

Within the diversified Web 2.0 technology landscape, a recent trend of social computing applications concerned the virtualization of human-based intellectual activities as electronic services. The business motivation for these developments can be identified as a combination of two orthogonal economic strategies: human capital outsourcing and business process rationalization. Human capital outsourcing offers benefits of increasing labor capacity and human potential in an organization without the indirect costs and risks that accompany long term employment relationships. Business process rationalization aims at optimizing and automating sourcing interdependent activities.

While rationalization is a classical domain of information systems, the outsourcing of labor witnesses enormous changes through the advent of modern Web technologies. On an individual basis, connectivity and interoperability of Web-based software applications fostered an increase of *homeshoring* so called *telecommuters* that offer their services to independently carry out a wide array of IT-enabled jobs via outsourcing portals like Odesk (`http://www.Odesk.com/`) or Elance (`http://www.Elance.com`). At the same time, the phenomenon of *wikinomics* opened up new possibilities of mass collaborations that allowed to drastically increase the scale of outsourcing labor while simultaneously decreasing activities into specialized microjobs. Respective *crowdsourcing* generally involves an organizational requester issuing calls to members of the public that carry out activities on individual or group basis [2]. Despite its potentials it has to be said that this approach also involves severe risks. Companies easily underestimate the challenges of managing crowdsourced projects that include issues like coordination, motivation, contracting, sustainability and loyalty. Another dimension of issues revolve around the virtual absence of employment rights.

For companies, the next step is to leverage wikinomics to a business context, where competition requires not only effectiveness but efficiency. This corresponds to a sharper focus on qualitative properties of functional business activities. For example, development of acceptable Wikipedia entries might take an unconstrained duration and is driven by public volunteers for idealistic reasons. In contrast, the documentation of product features in the crowdsourced development of a product manual needs to be finished in a given time and to a required standard in return of a monetary compensation in order to provide business value for the requester. Theoretical foundations for the qualitative optimization of massively collaborative processes might be found in areas like game theory or cybernetics. Their application however requires infrastructures for large scale

human interaction management in the context of mass collaboration. Such infrastructures are increasingly provided on the basis of electronic services that represent microjobs in crowdsourcing contexts and allow for their systematic and automatic sourcing and coordination. We refer to these as *people services* (pServices) and define them as Web based software services that deliver human intelligence, perception, or action to customers as massively scalable resources.

An early example of pService infrastructure is Amazon's Mturk platform that focuses on pService sourcing and provides a simple request-driven marketplace for so called human intelligence tasks (HITs). Mturk gained public attention during its application for searching half a million of satellite images for the lost vessel of Microsoft's Jim Gray to thousands of volunteers on the Web [3]. The failure of this undertaking might be considered an indication for a number of problems and open research questions that still exist for people service infrastructures. While problems like misunderstanding of requests and missing information on the results of others indicate shortcomings of pService sourcing and coordination, the fact that the mission was just not finished in time underlines the need to actively manage the quality of people services.

Building on pServices, organizational requesters might plan and operate massively distributed business processes by mapping their operational business models onto service-oriented architectures and platforms that are enabled for crowdsourcing. Such architectures and platforms leverage basic software service technologies like WS Human Tasks [4] that supports the implementation of human-based Web Services. However, the distinctive nature of crowdsourced business activities requires for rethinking the upper layers of the software service stack. These traditionally focus on workflows of classic business processes in a context of closed organizational models. Situations, where thousands of tasks need to be allocated to participants on the Web are so far not considered.

3 Requirements Analysis

Corresponding to the business goals of enabling crowdsourced business activities with qualitative optimizations, pService platforms have to provide means for two major classes of tasks: sourcing and coordination. pService sourcing leverages software service brokerage and marketplaces in order to discover and aggregate temporary workforce. pService coordination leverages software service mashups and composition in order to plan and enforce collaboration of temporary pService workers. An important aspect for pService platforms is to support service workers to understand pService semantics and provide ergonomic interfaces that incorporate mobile devices [5]. Additionally, a crucial point is for the platform methods to jointly provide a certain level of quality as required by organizational requesters. Only if qualitative properties like performance, scalability, availability, and correctness in relation to prices are observable and guaranteed, organizations will consider to incorporate pServices. Therefore pService sourcing not only needs to map pService requests to respective offers but also to ensure timely allocation with acceptable costs. Likewise, pService coordination not only

needs to regulate and enforce relationships of multiple pService tasks but also to introduce feedback loops for controlling correctness of results. The following list summarizes key quality requirements for pServices.

- *Performance:* Ability to return a result within certain response time limits
- *Scalability:* Ability to handle a certain average and peak number of requests
- *Availability:* Ability to continuously provide the service
- *Correctness:* Ability to return a minimum percentage of correct results

These quality requirements can typically not be met by individual service workers. Neither do these 'scale', nor are they necessarily available at any time requested or at once. Moreover, individual service workers won't be able to deliver correct results for all service requests. Thus, it seems inappropriate for service requesters to directly deal with individual service workers. Instead, their work should be managed by a pService platform that takes care of quality-of-service goals. Subsequently, we differentiate 2 types of pServices: individual ones performed by service workers and the managed ones requested by the service requesters for which SLOs can be defined and which might build on combinations of individual ones. We call the latter *managed people services* and define them as pServices that support service level objectives (SLOs) to guarantee certain quality-of-service goals like performance, scalability, availability, and correctness.

4 Platform Considerations

We see a need for people service platforms to support SLOs and guarantee QoS to service requesters. In the following, we provide some preliminary considerations about the functionality of such a platform. Fig. 1 describes the basic components.

Similar to Mturk, the proposed platform deals with two groups of customers: service requesters, who are submitting pService requests to the platform and service workers, who are willing to work on those requests. However, in our case, the pService requests are not directly passed to the service workers, but the platform deals with two different types of pService requests. *Managed pService requests* which are passed to the platform by the service requesters and *(native) pService requests* which are passed to the service workers by the platform.

A managed pService request consists of two parts: a service template and a set of SLOs. The service template refers to what is presented to service workers when delivering services. It includes informal and possibly semantic descriptions [6] of the problem to be solved as well as a response form to interact with service workers when delivering results. The SLOs comprise information about response time goals, scalability and availability requirements as well as the correctness of the results. Availability goals include the initial availability of the service (date, time), as well as the period of time it will be available for. Service workers can register at the platform and provide information about their skills, availability, and expected compensation with respect to service template classes.

Once a managed pService request is submitted to the platform, the planning component analyses the request. This includes forecasting whether enough service workers with appropriate skills will be available. In addition, an execution

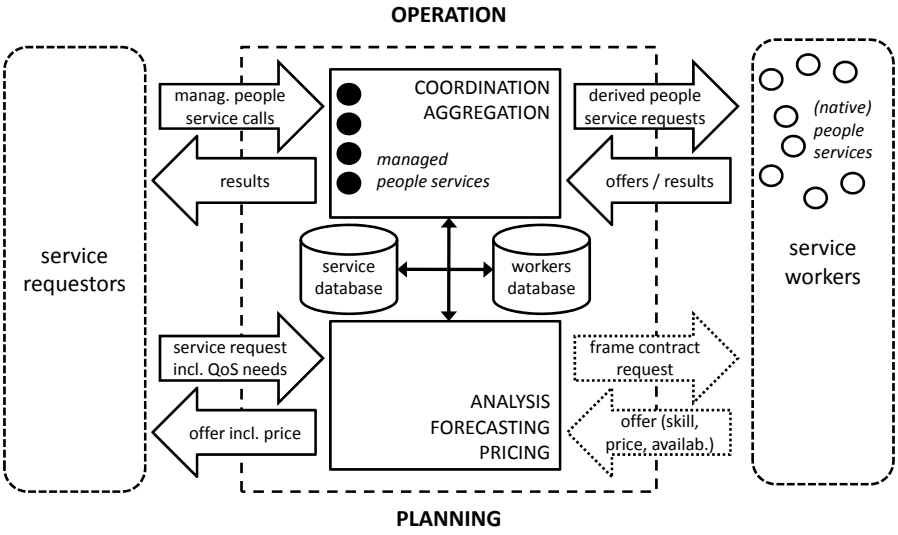

Fig. 1. Schematic view of proposed pServices platform

plan is created that addresses the service level objectives. Based on the type of service and the service level objectives, the platform generates a price offer for delivering the service to the requester as a managed pService.

At runtime, the service requester passes individual service calls to the platform. The platform operation engine will autonomously coordinate those service calls based on the execution plan which might make use of a variety of approaches to actively manage the SLOs: An incoming service call might be turned into multiple pService calls passed to service workers and results returned by one service worker might be compared with results from other service workers or sent to other service workers for validation. The quality of a service worker's contributions might be tracked with a reputation mechanism and might be used for quality forecasting. A notification mechanism might be used to actively pass requests to individual service workers to reduce lead time before they start working on a service. The same request might be sent to multiple service workers in parallel to ensure that at least one of them meets the response time goals. Variable incentives might be used to drive the service worker's motivation to meet response time goals, to be available even at unusual hours, and to deliver correct results. These and possibly more options are offered by the platforms as patterns that might be utilized in order to provide certain qualities of pServices. Many of those patterns build on the coordination of interactions between service workers in order to realize redundancy as well as feedback control loops. After completing the execution plan, the individual responses returned from the service workers are aggregated and returned to the service requester as a single result.

The business model of the platform is based on the gap between the fixed price for the managed pServices paid by the service requester and the average

incentives paid to the service workers. Based on information about registered service workers and historical information, the future incentives might be estimated and used for pricing of new pServices.

5 Summary and Outlook

There are two general approaches for incorporating human activities into electronic business processes or applications, each having a major deficiency. Traditional human tasks of business processes lack scalability because they are typically performed by a closed group of people. Recent approaches like Amazon's Mturk are addressing the scalability issue by dealing with a virtually unlimited number of service workers over the Internet but at the same time lack quality because of limited control of the workforce.

After introducing the term of people service (pService) for a broad type of electronic services that leverage human intelligence, perception, or action, this paper discussed the need for pServices to meet quality-of-service goals. An integrated platform and business model was outlined, which is capable of supporting SLOs for pServices by actively managing contributions of service workers.

Additional research is required for many aspects of the platform, e.g. the formal description of the service requests which allow for matching them to the skill profiles of service workers, reputation mechanisms and incentive models for motivating the workers to do their best, and coordination patterns to produce high quality results based on the contributions of multiple service workers. We plan to substantiate our concept by implementing a prototype for the proposed pService platform and validate the concept based on different usage scenarios.

References

1. Pontin, J.: Artificial intelligence, with help from the humans. The New York Times `http://tinyurl.com/58nec5` (25 March 2007)
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) (2008) 75–90
3. Silberman, S.: Inside the high-tech search for a silicon valley legend. Wired magazine `http://tinyurl.com/358k7f` (24 July 2007)
4. Agrawal, A., Amend, M., Das, M., et al.: Web Services Human Task (WS-HumanTask), Version 1.0. Technical report, Active Endpoints Inc., Adobe Systems Inc., BEA Systems Inc., IBM Corporation, Oracle Inc., and SAP AG (2007)
5. Maleshkova, M., Komazec, S., Grasic, B., Denaux, R.: iService: Human Computation through Semantic Web Services. Future Internet Symposium (FIS 2008), http://tinyurl.com/5fr3po (October 1st 2008)
6. Sycara, K., Paolucci, M., Ankolekar, A., Srinivasan, N.: Automated Discovery, Interaction and Composition of Semantic Web Services. Journal of Web Semantics **1**(1) (December 2003) 27–46