Information Systems ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at SciVerse ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

Auction-based crowdsourcing supporting skill management

Benjamin Satzger [a,*], Harald Psaier [a], Daniel Schall [b], Schahram Dustdar [a]

[a] Distributed Systems Group, Vienna University of Technology, Argentinierstrasse 8/184-1, A-1040 Vienna, Austria
[b] Siemens Corporate Technology (CT T CEE), Siemensstrasse 90, 1211 Wien, Austria

ARTICLE INFO

Keywords:
Human-centric BPM
Crowdsourcing
Online communities
Task markets
Auctions
Skill evolution

ABSTRACT

Crowdsourcing is a promising approach for enterprises to maintain a flexible workforce that is able to solve parts of business processes formerly processed in-house. Companies perceive crowdsourcing as a concept that allows receiving solutions quickly and at little cost. Similar to cloud computing where computing power is provided on demand, the crowd promises a flexible on-demand workforce. However, businesses realize that these benefits entail a lack of quality control. The main difference compared to traditional approaches in business process execution is that tasks or activities cannot be directly assigned to employees but are posted to the crowdsourcing platform. Its members can choose deliberately which tasks to book and work on. In fact, crowdsourcing is heavily affected by the loose-coupling of workers to crowdsourcers and the dynamics of the environment. Hence, it remains a major challenge to guarantee high-quality processing of tasks within the prescribed time limit. A further obstacle for adoption of crowdsourcing in enterprises is the fact that it is hard to specify a fair monetary reward in advance. The concepts introduced in this work allow to smoothly integrate new workers, to keep them motivated, and to help them develop and improve skills needed in the system. We present a crowdsourcing marketplace that matches complex tasks, requiring multiple skills, to suitable workers. The key to ensuring high quality lies in skilled members whose capabilities can be estimated correctly. To that end, we present auction mechanisms that help to correctly estimate workers and to evolve skills that are needed in the system. Crowdsourcers do not need to predefine exact prices but only maximum prices they are willing to pay since the actual rewards for tasks are formed by supply and demand. Extensive experiments show that our approach leads to improved crowdsourcing, in most cases.

1. Introduction

Crowdsourcing is a new paradigm for performing computations in Web-based environments by utilizing the capabilities of human workers. The idea of crowdsourcing is sometimes referred to as *human computation*, a methodology that lets humans process tasks which are difficult to implement in software. Such tasks include transcription of documents, reviewing of articles or evaluating the quality of ranking algorithms.

A major problem with current crowdsourcing environments as described in [1,2] is the lack of manageability as a result of the openness of Web based platforms, where anybody can join and participate. Whilst this openness, which allows to reach out to and attract members with different knowledge and interests, is an advantage of crowdsourcing, it is at the same time the reason that makes quality assurance particularly challenging. The crowdsourcing trend attributes partly to the success of outsourcing in general. With ever changing requirements,

* Corresponding author. Tel.: +43 15880118418.
E-mail addresses: satzger@dsg.tuwien.ac.at (B. Satzger), psaier@dsg.tuwien.ac.at (H. Psaier), daniel.schall@siemens.com (D. Schall), dustdar@dsg.tuwien.ac.at (S. Dustdar).

0306-4379/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.is.2012.09.003

in-house business processes need to adapt to changing situations rapidly in order to stay competitive. Often, changes involve not only the need for process adaptation, but also, require an additional inclusion of new capabilities and knowledge, previously unavailable to the company. Hence, outsourcing of parts of business processes are an attractive model. Today, quality control in crowdsourcing often comes down to manually submit identical jobs multiple times to a crowd of workers. This, however, is only feasible for simple low-priced tasks and is in conflict with highly automated business process execution.

Apart from the fact that only very simple tasks are crowdsourced today, there is another major difference between crowdsourcing and task delegation in enterprise environments: the mapping of tasks to workers processing them. In enterprises tasks are usually directly assigned to employees while in crowdsourcing tasks are booked voluntarily. Workers are loosely coupled to the crowdsourcing platform compared to employees, and consist of heterogeneous members with different interests, working style, cultural background, and skills. Workers may join and leave at any time. This heterogeneity and freedom make it hard to ensure quality. A crowdsourcing task usually consists of a textual description and some reward. Workers can browse online through the list of published tasks and choose the ones they like. For crowdsourcers it is very hard to determine a reward that motivates workers to book the task but is not too high, either. This does not only depend on the task itself but also on external factors, e.g., number of available workers and the workers' context including social environment and environmental influences, like weather.

Nevertheless, crowdsourcing is pushed by large IT companies such as Amazon, Google, or Yahoo!. They have recognized the opportunities behind such *mass collaboration systems* [3] for both improving their own services and as business case. In particular, Amazon focuses on a task-based crowdsourcing platform called *Amazon Mechanical Turk* (AMT) [4]. Requesters are invited to issue *human-intelligence tasks* (HITs) requiring a certain qualification to the AMT. These crowdsourcers post mostly simple tasks that, however, require human capabilities. In particular, 50% of tasks are processed at a cost of $0.10 and less, most of the tasks are usually also offered in chunks to multiple AMT workers [5].

Our approach helps to establish crowdsourcing in a business environment. There are already providers that target at enterprise crowdsourcing, such as CrowdFlower [6] who broker crowd resources to customers to overcome quality and reliability issues. However, automated quality assurance and managing and adapting the crowd in an automated manner remains challenging. Crowd customers prefer fully automated deployment of their tasks to a crowd, just as in common business process models. In this paper, we propose a solution suitable in combination with the service-oriented architecture (SOA) paradigm. SOAs are an ideal grounding for distributed environments. With their notion of the participants as services and registries, resources can be easily and even automatically discovered for composing whole business processes. A plethora of standards supports seamless integration and registration of new services, and provides protocols for communication, interaction and control of the components. Altogether, we believe SOAs together with automated crowdsourcing of tasks provide an intuitive and convenient technical grounding to automate large-scale crowdsourcing environments. Important to note, today SOA not only includes software-based services, but also *Human-Provided Services* [7] and *BPEL4People* [8] for human interactions in business processes and allow to express mass collaboration environments.

This paper proposes to use auctions to map tasks to workers in the crowd. Thus, rewards for tasks are built based on supply and demand; crowdsourcers do not have to "guess" a fair, competitive reward, but may define a maximum amount of money they are willing to pay. This prevents from overpaying or underpaying; the latter may be equally bad and result in "non-sellers" with the potential to cause delays, e.g., in the context of business process execution. The auctioning mechanism does not blindly look at prices only, but includes several techniques to ensure quality. In addition to the pure mapping of tasks to workers we address how to build and manage an automated crowdsourcing platform. For establishing a successful crowdsourcing environment it is important to maintain a motivated base of crowd members and provide stimulus for learning required skills. Only a recurring, satisfied crowd staff is able to ensure high quality and high output. A *skill evolution* model supports new and existing crowd workers in developing capabilities and knowledge needed by crowdsourcers. All standard processes in the crowdsourcing platform are automated and free from intervention, which allows to handle a vast amount of tasks and makes it compatible with a SOA approach. To ensure sustainability, the model is designed to not only maximize the benefit of crowdsourcers, but also takes the welfare of workers into account. The main contributions of this work are:

- *Automated matching and auctions.* For providing a beneficial distribution of tasks to the available resources we organize auctions taking into account price and the suitability of workers estimated based on generated user profiles.
- *Stimulating skill evolution.* In order to bootstrap new skills and unexperienced workers we provide skill evolution by integrating assessment tasks into our auction model.
- *Evaluation.* Extensive experiments covering various scenarios quantify the advantages of a skill evolution based approach in comparison to traditional auctions.

This paper extends previous work presented in [9] mainly by providing a much more detailed discussion of our approach including relations to other work. The evaluation focus of the previous work is on a crowdsourcing marketplace with only one relevant skill, which refers to a highly specialized platform. Here, we specifically focus on environments where multiple skills need to be taken into account, as necessary in more general, business-oriented crowdsourcing.

Please cite this article as: B. Satzger, et al., Auction-based crowdsourcing supporting skill management, Information Systems (2012), http://dx.doi.org/10.1016/j.is.2012.09.003

In most enterprise settings tasks are relatively complex and multiple skills are necessary for providing successful solutions. Due to less rigid space constraints we are also able to present our results in more depth than before.

The remainder of this paper is structured as follows: In Section 2 related work is discussed. Section 3 describes the design of our crowdsourcing system, including its actors and their interaction. Then, Section 4 details our adaptive auction mechanisms and Section 5 presents the conducted experiments and discusses their results. Section 6 concludes the paper and points to future work.

2. Related work

This section looks at crowdsourcing from various angles, i.e., the role of crowdsourcing in socially enhanced SOA, existing platforms, assignment of tasks in crowdsourcing based on incentives, quality assurance in crowdsourcing. A short introduction into auction mechanisms is given before some open challenges in crowdsourcing are pointed out.

2.1. Crowdsourcing and socially enhanced SOA

SOA, and in particular Web services, play a fundamental role in supporting flexible, cross-enterprise collaboration scenarios. In addition, several standards, specifications and models render Web-services a convenient foundation for designing, deploying, but also monitoring and adapting dynamic service-oriented environments. For example, the previously mentioned *Human-Provided Services* (*HPS*) model [7] enhances the traditional SOA-based systems by enabling people to provide services with the very same technology used by implementations of traditional software-based services (SBS). A HPS interface allows humans defining and providing their service transparently. Across this particular interface they are able to participate in ad hoc as well as process-centric collaborations. The approaches presented in [10,11] deal with the execution of business processes in such a setting. In this work we position crowdsourcing in a service-oriented business setting by providing automation. In crowdsourcing environments, people offer their skills and capabilities in a service-oriented manner. Major industry players have been working towards standardized protocols and languages for interfacing with people in SOA. Specifications such as WS-HumanTask [12] and BPEL4People [8] have been defined to address the lack of human interactions in service-oriented businesses [13]. These standards, however, have been designed to model interactions in closed enterprise environments where people have predefined, mostly static, roles and responsibilities. Here we address the service-oriented integration of human capabilities situated in a much more dynamic environment where the availability of people is under constant flux and change [14].

2.2. Crowdsourcing platforms

Today, a plethora of different mass collaboration system and crowdsourcing platforms can be found and classified into several categories [3,15]. Crowdsourcing is often defined very broadly, and many regard Wikipedia and Linux as famous examples of crowdsourcing. In this work we focus on crowdsourcing platforms that distribute *tasks* to a mass of users. In the following we discuss two representatives, Amazon Mechanical Turk (AMT) and Yahoo Answers! (YA), more closely and detail their task distribution process including assignment and quality assurance strategies.

YA, is a question and answer portal, mainly based on interactions between members. Tasks are questions which are asked and answered by members. The interesting aspect of YA relevant for our crowdsourcing approach is the role of *two-sided markets* [16]. In YA, users get 100 points by signing-up to the platform [17]. For each answer being provided, users get additional points (more points if the answer is selected as best answer). However, users get negative points if they ask questions, thereby encouraging members to provide answers. Based on the rewarding scheme in YA, users tend to have both roles instead of either being answerer (worker) or asker (crowdsourcer, requester). In the context of YA and human-reviewed data, [18] provides an analysis of data quality, throughput and user behavior.

AMT is a platform that offers access to the largest number of crowdsourcing workers and tasks, both in number and diversity of topic. In March 2012, more than 250,000 workers were registered and over 280,000 tasks were available. Already in 2010, about 50,000-100,000 tasks were available at any given time [19]. With their notion of HITs (Human Intelligence Tasks) that can be created using a Web service-based interface they are closely related to our aim of mediating the capabilities of crowds to service-oriented business environments. According to one of the latest analysis of AMT [5], HIT topics include typical activities which are still more precisely solved by humans than machines. Examples include transcription, classification, and categorizations tasks for documents and images. Furthermore, there is also tasks for collecting data, image tagging, and feedback or advice on different subjects. In the following subsections we examine task handling and related issues in such environments in greater detail.

2.3. Crowdsourcing assignment and incentives

A main difference of crowdsourcing compared to most other forms of task delegation in enterprises is that crowd members voluntarily agree to solve crowdsourced problems motivated by incentives, such as money or social recognition. While traditional employment relationships constitute authorization to issue directives to subordinates, in crowdsourcing the workers are loosely bound, non-hierarchical, and self-motivated. This freedom increases the attractiveness for crowd workers and allows users to organize work according to their individual habits regarding, e.g., working hours. This loose involvement of the workers confines the management scope.

Most task-based crowdsourcing platforms provide requesters the possibility to publish simple task descriptions into a database all workers have access to. A task description in AMT, for instance, consists of a title, textual description, expiration date, time allotted, keywords,

Please cite this article as: B. Satzger, et al., Auction-based crowdsourcing supporting skill management, Information Systems (2012), http://dx.doi.org/10.1016/j.is.2012.09.003

required qualifications, and monetary reward. According to [5], in AMT 10% of the offered tasks include a payment of less than or equal to $0.02, 50% of the HITs have a price above $0.10 and 15% of $1 or more. The prices are defined by the crowdsourcers. For a enterprise crowdsourcing platform designed for complex task of high value the pricing is much more important and difficult. Our auctioning approach relieves crowdsourcers from defining the monetary reward in advance; prices are formed according to supply and demand but requesters can define a maximum amount of money they are willing to pay.

The AMT website allows workers to search for tasks using a predefined set of criteria [5]. Despite this support, finding appealing tasks can be cumbersome [20]. In this paper we propose to invite all promising workers to submit a bid that have the ability to solve a task. This reasoning is based on their track record. Workers do not have to browse through a large list of tasks but are only exposed to tasks they have a realistic chance to process successfully.

2.4. Crowdsourcing quality assurance

Loose coupling, individual availability of workers and low prices contribute to issues with quality assurance. This has already been observed by researchers which provided approaches like EM-based algorithms for scoring the bias of workers [21], Bayesian algorithms that correct the workers bias and increases accuracy with redundant workers, or classification for identifying low-quality workers. The authors of [22] note that these research approaches consider a major redundancy of workers (at least 10 workers per task) to achieve their quality goals. Also, they point out that for particular categorization tasks humans are still less reliable than modern machine learning techniques. The lack of quality control has also been recognized by existing platform customers and some now provide their individual business solutions. Generally, these offer a mediation service between crowd customers and the workers [6,23]. Furthermore, there is platforms that specialize on particular task types (e.g., [24]) and try to develop and maintain a community of trusted workers to guarantee quality results. In our previous work [10], we extend BPEL4People to leverage social networks for business process execution. Quality is supported by roles and ranking algorithms for assigning them to users in the social network. In [25] we present similarity metrics for task requirements and skill matching related to cosine similarity.

2.5. Auctions

Auctions are a very old idea already used by the Babylonians but still an active area of research. The rise of e-commerce has drastically increased the number and diversity of goods traded via auctions. Many recently installed markets, such as energy or pollution permit markets, are based on auctions [26]. There are many different flavors of auctions differing in the number of items considered (single/multi-item), the number of buyers and sellers (demand/supply/double auction), the bidding procedure (open/closed bids and ascending/descending), and how the price is determined (e.g., first/second price); however, four standard types are widely used [26]. They all assume a single seller and multiple buyers (demand auction) and, in their simplest forms, a single item to sell (single-item auction). The so-called English auction is an ascending open-bid auction, the Dutch auction is a descending open-bid auction. That is, in the English auction, the price is raised until only one buyer is left; the Dutch auction starts at a very high price, which is continuously lowered. The first buyer who accepts the price wins the auction. The other two standard auction types are closed-bid auctions, i.e., each bidder submits a single bid which is hidden for other buyers. We use an adapted version of a closed-bid auction; a single auction deals with the matching of one task to one or many crowd workers (single-item demand auction). Research at the intersection of crowdsourcing and auctions so far investigates only all-pay auctions [27–29]. All-pay auctions [30] are auctions in which all bidders must pay independent of whether they win the item; the highest bidder wins the auction. This form of auction models the case of a crowdsourcing contest like a design contest, in which all designers "pay" with their effort, but only the best design is rewarded. However, in an enterprise setting, such a model is not suitable. Ref. [31] states that industrial procurement auctions are almost always closed-bid auction. Since the procurement setting shares similar characteristics to task-based crowdsourcing and is accepted in business environments we also choose closed-bid auctions. Another reason for choosing closed-bid auctions is that we determine the winning bid not only on the price but also on the workers' skills. With open bids it would be obvious for everyone that the worker with the best (lowest) price is less qualified than the winner with a higher price.

2.6. Open challenges in crowdsourcing

Platforms and research in the area of crowdsourcing are gaining momentum. Currently, crowdsourcing platforms employ simple mechanism for quality assurance, e.g., crowdsourcers rewarding good work with bonuses or refuse payment for poor performance. Requesters can also block a particular worker from completing future tasks. Conversely, workers can filter out requesters who fail to provide sufficient justification for rejecting a performed task [32]. Rewards need to be set explicitly by requesters for every tasks. Based on the overview of related work given above we think that the state of the art still lacks approaches for automated mapping of tasks to the crowd considering quality assurance, incentives, and maintaining a powerful workforce. In this work we address exactly these issues by proposing a marketplace based on auctions. We extend traditional auction mechanisms to incorporate automated quality assurance and skill management for workers.

3. Design of marketplaces in crowdsourcing

The core activity in task-based crowd environments is members providing their labor by processing tasks. In this

Please cite this article as: B. Satzger, et al., Auction-based crowdsourcing supporting skill management, Information Systems (2012), http://dx.doi.org/10.1016/j.is.2012.09.003

section, we explain our idea of task-based crowdsourcing on a market-oriented platform. The aim is to organize and manage the platform to the satisfaction and benefit of all participants; crowd members and platform provider. We will now introduce the basic design of the proposed crowdsourcing environment.

In task markets different stakeholders can be identified. Generally, there is the requesters and workers representing the registered members of a crowd marketplace. The task of the third stakeholder, the crowd operator in between, is to manage the crowd task auctions. To satisfy any of the stakeholders the operator must assure that the requesters obtain a result of high quality in a timely manner. On the other hand, the workers would like to have tasks available whenever they are motivated to work and are interested in a high reward for processing a task. The operator itself works towards a long-term profit. To bootstrap the skill-based system, each member interested in offering of processing tasks is required to create a profile containing information about her/his interests and skills. The basic interactions and an external view on the proposed crowdsourcing environment are depicted in Fig. 1.

The crowdsourcing environment consists of members who can participate in transactions (see Fig. 1(a)). Within a particular transaction a member can either adopt the role of a requester **R**, who initiates a transaction by announcing tasks (see Fig. 1(b)), or the role of a worker **W**, who processes a task. We propose a crowdsourcing marketplace that handles transactions transparently for its members; requesters and workers do not communicate directly, but only with the crowdsourcing marketplace in between. We argue that this standardized style of interaction is less prone for misconceptions and more efficient because it allows members getting used to the system. Tasks (Fig. 1(b)) are created by requesters based on their current needs. Requesters initiate a transaction by submitting a task to the marketplace, with additional information about the amount of money he is willing to pay for the processing of the task and additional requirements (Fig. 1(c)). It is the responsibility of the marketplace operator to find a suitable worker, to submit the task to the worker, to collect the result, and to transmit it to the requester.

The interaction of a worker with the market platform is initiated by the latter by asking a member whether s/he is interested in processing a task (Fig. 1(d)). This interest can be expressed by bidding for the task. Workers have skill profiles denoted by the symbol **P**. These profiles are not statically defined, but are updated based on the level of delivered task quality. This procedure ensures an up-to-date view on workers' capabilities and skills. Based on the bids and background information about the bidders, the system selects one or multiple workers, who are then asked to process the task.

4. Auction-based task assignment

In the following we discuss the steps involved in transaction processing and outline the novel idea of skill evolution.

4.1. Processing of transactions

Fig. 2 illustrates the steps involved in the internal processing of a transaction. In the qualification step the marketplace identifies all members capable of processing the task (Fig. 2(a)), based on the task description and the members' profiles. The preselection chooses a limited number of the most suitable workers (Fig. 2(b)) to have a reasonable amount of participants for the auction. The preselection step helps to avoid a flooding of auction announcements. In this way members are only exposed to tasks/auctions that match their skills and for which they actually have a realistic chance of being accepted as worker. Due to the voluntary, open nature of crowdsourcing environments, not all preselected workers may decide to follow the invitation to compete in an auction.

This fact is depicted by the transition phase between Fig. 2(b) and (c) where only a subset of preselected workers decides to participate. The auction phase (Fig. 2(c)) allows

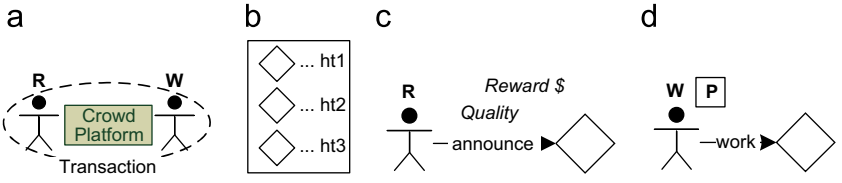


Fig. 1. Crowd environment building blocks and interaction of stakeholders: (a) crowd; (b) tasks; (c) requester; (d) worker.

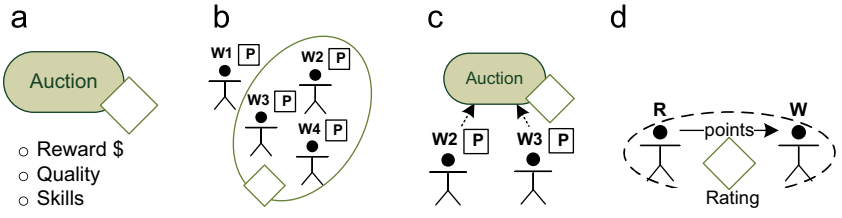


Fig. 2. Internal processing of a transaction: (a) auction; (b) preselection; (c) biddings; (d) feedback.

Please cite this article as: B. Satzger, et al., Auction-based crowdsourcing supporting skill management, Information Systems (2012), http://dx.doi.org/10.1016/j.is.2012.09.003

each participant to submit an offer and finally, a winner is determined who is supposed to process the task. In the case of a successful processing the marketplace returns the final result to the requester, handles the payment, and allows the requester to give feedback about the transaction in the form of a rating (Fig. 2(d)).

As mentioned before, tasks come with a description, a maximum amount of money the requester is willing to pay and further requirements, i.e., time requirements and quality requirements. The former is typically given in the form of a deadline, the latter could range from a simple categorization (e.g., low, high) to sophisticated quality requirement models. Each worker has a self-provided profile describing her/his skills and, additionally, the marketplace operator is keeping track of the actual performance. We propose to maintain a performance value per user and skill, encoded as tuple consisting of the observed performance and the confidence in that value. The input used to generate these performance values comes from the ratings of the requesters and a check whether the deadline was met, which can be performed by the system without feedback from the requester. The qualification phase is based on a matching of the task description to the skills of the members considering their performance and confidence values. Higher requirements impose higher standards on the performance of the member. The result of this matching is a boolean value indicating whether a member is meeting the minimum requirements. In the next step, the preselection, the qualified members are ranked based on skill, performance, and the confidence in the performance; only the top-k members are chosen to participate in the auction. This helps to reduce the number of auction requests to members in order to avoid spamming members and to spare members the frustration caused by not winning the auction. The marketplace operator as the auctioneer hides parts of the task's data. Workers only see the task description and the time and quality requirements, but not the associated price determined by the requester. The auction is performed as a closed bid auction, whereas each participant is only allowed one bid. At the end of the auction a winner is determined based on the amounts of the bids and the performance–confidence combination of the bidders' skills. If all bids are higher than the amount the requester is willing to pay the auctioneer would typically reject all bids and inform the requester that the task cannot be assigned under the current conditions. In this case the requester could change the task by increasing the earnings, lowering the quality requirements or extending the deadline and resubmit the task. With a selection strategy outlined in more detail in the next section the marketplace assigns the task to the worker for processing. After the processing of the task by the worker and the receipt of rating information, the performance of the worker is adjusted and the confidence value is increased.

Technically, an aptitude function estimates how well workers are suited for handling a task. It is used as basis for qualification and preselection and can be formally defined as

$$aptitude : W \times T \rightarrow [0,1], \quad (1)$$

where W is the set of workers and T represents tasks. $aptitude(w,t) = 1$ would mean that worker $w \in W$ is perfectly qualified for handling task $t \in T$. A mapping to zero would represent a total inaptness. Similarly, a ranking function is used to rank workers' bids:

$$rank : W \times T \times B \rightarrow [0,1], \quad (2)$$

where B is the set of bids. In addition to the aptitude, the *rank* function also takes monetary aspects, contained in bid $b \in B$, into account. A property of a sound ranking function is that if two workers have the same aptitude for a task then the one with the lower bid will have a higher rank. The *aptitude* function is used for performing qualification and preselection. As auction admittance strategy you can either admit all workers with an aptitude higher than a certain threshold, the top-k workers according to aptitude, or a combination of the two strategies. The ranking function is used to determine the winner of an auction: the highest ranked worker.

4.2. Skill evolution

The concepts discussed so far provide the fundamentals for automated matching of tasks to workers. As outlined previously, a further major challenge hampering the establishment of a new service-oriented computing paradigm spanning enterprise and open crowdsourcing environments are quality issues. In our scenario this is strongly connected to correctly estimating the skills of workers. One approach for increasing the confidence in worker skills are qualification tasks, with the shortcoming that these tasks would need to be created (manually) by the requesters who have the necessary knowledge. This implies a huge overhead for the testing requester; s/he is also the only one who benefits from the gathered insights. Here, we take a different approach by integrating the capability of confidence management into the crowdsourcing platform itself. Instead of having point-to-point tests, we propose the automated assessment of workers to unburden requesters in inspecting workers' skills. We believe that this approach offers great potential for the (semi-)automatic inclusion of crowd capabilities in business environments. The first challenge one needs to address is to cope with the "hostile" environment in which computing is performed. Workers may cheat on results (e.g., copy and paste of existing results available in the platform), spam the platform with unusable task results, or even provide false information. A well-known principle in open, Web-based communities is the notion of *authoritative* sources that act as points of references. For example, this principle has been applied on the Web to propagate trust based on *good seeds*. Our idea of skill evolution is in a manner similar. We propose the *automatic assessment* of workers where confidence values are low. For example, newcomers who recently signed up to the platform may be high or low performers. To unveil the tendency of a worker, we create a hidden "tandem" task assignment comprising a worker whose skills are known (high performer) with a high confidence and a worker where the crowdsourcing platform has limited knowledge about its skills (i.e., low confidence). The next step is that

Please cite this article as: B. Satzger, et al., Auction-based crowdsourcing supporting skill management, Information Systems (2012), http://dx.doi.org/10.1016/j.is.2012.09.003

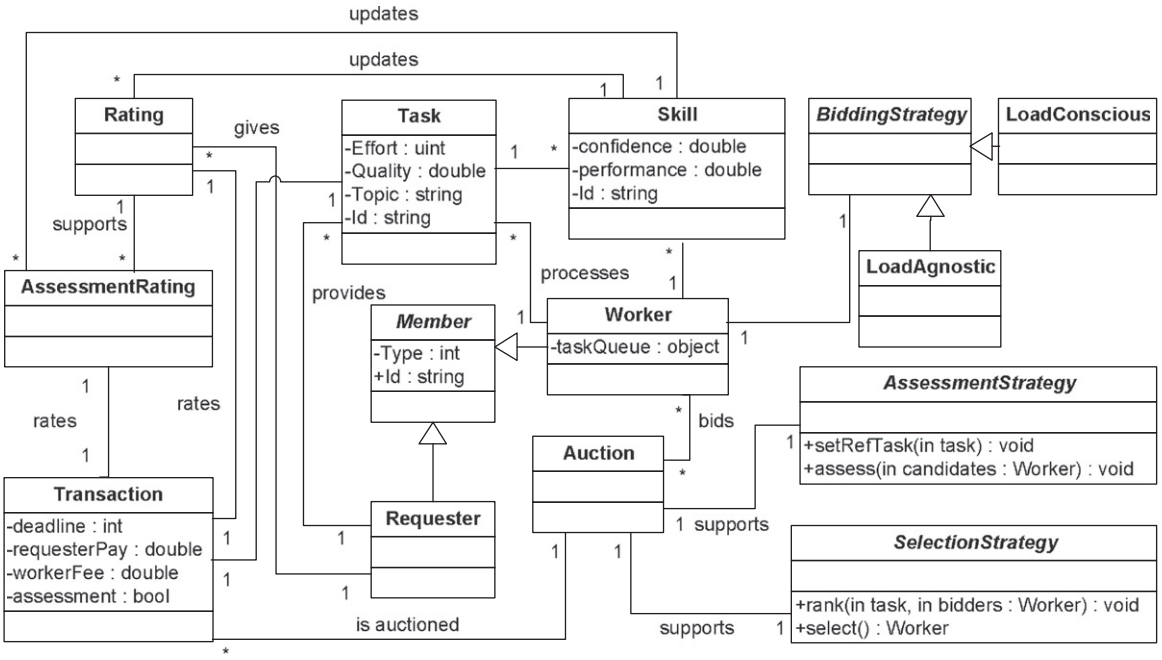


Fig. 3. Entity structure of the simulation environment.

both workers process the *same* task in the context of a requester's (real) task. However, only the result of the high confidence worker is returned to the requester, whereas the result of the low confidence worker is compared against the delivered reference. This approach has advantages and drawbacks. First, skill evolution through tandem assignments provides an elegant solution to avoid training tasks (assessments are created automatically and managed by the platform) and also implicitly stimulates a learning effect. Of course, the crowdsourcing platform cannot charge the requester for tandem task assignments since it mainly helps the platform to better understand the true skill (confidence) of a worker. Thus, the platform must pay for worker assessments. As we shall show later in our evaluation, performing assessments provides the positive effect that the overall quality of provided results and thus requester satisfaction increases due to a better understanding of worker skills. We embed skill evolution in our crowdsourcing platform as follows. After the winner of an auction has been determined it is evaluated whether an assessment task is issued to further workers. The function *assess* outputs 1 if an assessment task is to be assigned to a worker and 0 otherwise.

$$assess : W \times T \times B \times W \rightarrow \{0,1\}. \quad (3)$$

An input tuple (w,t,b,w_r) checks whether tasks $t \in T$ is to be assigned to $w \in W$ who offered bid $b \in B$. Worker $w_r \in W$ is the reference worker, in our case the worker who has won the corresponding auction and who will thus process the same task.

5. Implementation and evaluation

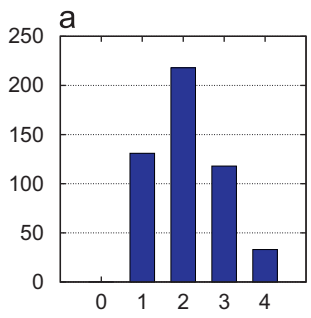
In this section we discuss implementation aspects, introduce the detailed design of our experiments and present results. We have implemented a Java-based simulation framework that supports all previously introduced concepts and interactions between requesters, the platform, and workers. All the above introduced functions (1)–(3) have been implemented in our framework. We describe the scenarios considered for evaluation using the framework. Then, the results of the evaluation are presented and discussed.

5.1. Simulation environment

Fig. 3 details the most important entities of our auction based crowdsourcing marketplace simulation environment and their dependencies. The interested reader can also try out a Web-based demo online.[1]

- *Member*. The crowd's uniquely identifiable Members are comprised Workers and Requesters.
- *Transaction*. The life-cycle of a Transaction instance begins once a requester decides to issue a Task to the platform.
- *Auction*. The Auction class is responsible for conducting auctions.
- *Selection Strategy*. Workers selected by a Selection Strategy may submit their bids for the announced auctions. Each worker has an individual *BiddingStrategy*.
- *AssessmentStrategy*. Once all bids are collected, the platform decides on the winner and assessment task using SelectionStrategy and AssessmentStrategy. Depending on the auction strategy, the same task is assigned to a bunch of workers for assessment.

[1] http://www.infosys.tuwien.ac.at/prototyp/Crowds/Markets_index.html



	Parameters
$pfmc_{real}$	$\{\mu$=0.25, μ=0.5, μ=0.75$\}$
$cnfd$	$\{\mu$=0.25, μ=0.5, μ=0.75$\}$
#requesters	{5, 50, 100}
skill evolution	{false, true}

Fig. 4. Parameters used for scenario generation (a) and distribution of workers skills (b): (a) worker skills; (b) scenarios.

- *Rating*. Completed tasks may be rated by requesters as implemented in Rating. However, only one task (the actually returned) may be rated by the worker.
- *AssessmentRating*. AssessmentRating is used for rating of assessment tasks. Finally, both types of rating update the skill profile of a worker.

5.2. Experiment design

Scenarios. An evaluation scenario consists of a set of workers W and a set of requesters R. In every round of the simulation each requester usually announces a task. An auction is conducted for each announced task t, which consists of a description of the skills needed for its processing, an expected duration, a deadline, and the expected quality. The description of the skills needed for a task mainly consists of a weight function:

$$weight : T \times S \rightarrow [0,1].$$

This function tells for each skill $s \in S$ how important it is to process task $t \in T$. If a skill s is not needed to process task t then $weight(t,s) = 0$.

High quality requirements for a task indicate highly sophisticated and demanding tasks. For each worker w and skill s the platform maintains a performance value $pfmc(w,s)$ and a confidence in that value $cnfd(w,s)$. This observed performance value is derived from requester ratings; if it is based on many ratings the confidence is close to one, if there are only a few ratings available the confidence is close to zero. Based on task t's skill requirements and a worker w's performance/confidence values for these skills it is possible to calculate the suitability of w for task t. For the evaluation we assume that each worker w has a certain performance $pfmc_{real}(w,s)$ for each skill $s \in S$ which is hidden but affects the quality of the results. Requesters rate the workers based on the results which in turn is the basis for the observed performance and confidence values. We assume that the processing of tasks demanding a certain skill causes a training effect of that skill, i.e., $pfmc_{real}(w,s)$ increases. In contrast to our previous work [9], where we made the simplifying assumption of having only one skill, in this work we explicitly consider multiple skills.

For the simulation we have created 500 workers assuming four different skills. Each worker has at least one skill and a maximum of four skills. For each worker and each of the four skills a coin is tossed to determine whether a worker has the respective skill. On average, this process leads to a skill distribution as illustrated in Fig. 4(a), which shows the number of workers (y-axis) having 0,1,2,3, or 4 skills (x-axis). The figure shows that, on average, the initialization creates a 40% majority of the workers with at least two skills. There are almost as many (around 25%) that have only 1 or 3 skills. A minority of workers has all four skills. There is no scenario with workers having no single skill.

Each worker is created with random values for each skill, i.e., random values for $pfmc_{real}(w,s)$ and $cnfd(w,s)$ are drawn according to a normal distribution $\mathcal{N}(\mu,\sigma^2)$, for each skill s of each worker w. The performance value $pfmc(w,s)$ is set according to the formula $pfmc_{real}(w,s)+\mathcal{N}(0,1-cnfd(w,s))$ which ensures that for high confidence the expected deviation of $pfmc(w,s)$ from $pfmc_{real}(w,s)$ is small and for low confidence values it is high, respectively. All values are restricted to the range of [0,1].

Fig. 4(b) describes the different scenarios used for evaluation by showing the different parameters used for generation of scenarios. Their permutation yields all possible variants of our simulation, i.e., we consider 54 different settings. The first row shows the different mean values used for generating $pfmc_{real}(w,s)$ for each worker and skill. Low values result in worker populations with low average skills while high values result in highly skilled workers. Similarly, the second row states parameters used to define the mean for the generation of the $conf(w,s)$ values. Low values refer to a worker base the system has only little knowledge about, i.e., the true skills can be estimated only vaguely; high values lead to systems which have profound knowledge about the workers' skills. Given the two generated values $pfmc_{real}(w,s)$ and $cnfd(w,s)$ the observed performance is randomly drawn according to $\mathcal{N}(pfmc_{real}(w,s),1-cnfd(w,s))$. Hence, a low confidence in the performance leads to highly distorted values for $pfmc(w,s)$, higher confidence values decrease the variance. The third row specifies the number of requesters and defines whether the system is experiencing high or low load. Finally, the last row defines whether the skill evolution is used or not. For all experiments averaged results are used; each single evaluation run consists of 1000 steps.

To illustrate the influence of the real performance and confidence on the created workers, Fig. 5 gives an overview of the statistical distributions in the form of

Please cite this article as: B. Satzger, et al., Auction-based crowdsourcing supporting skill management, Information Systems (2012), http://dx.doi.org/10.1016/j.is.2012.09.003

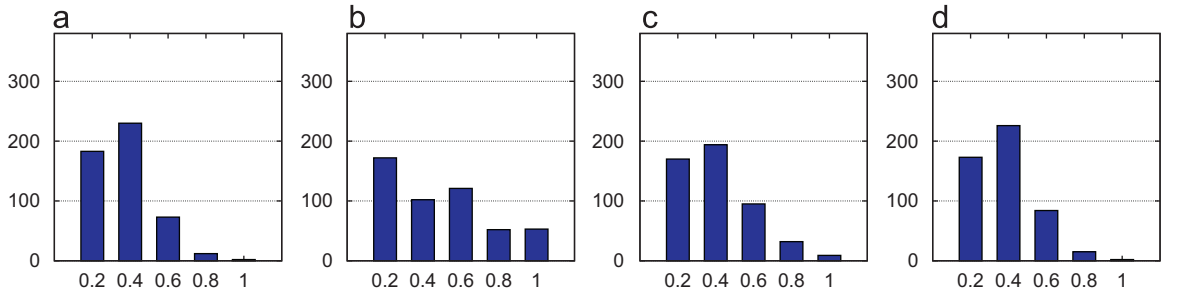


Fig. 5. Generated worker population according to a low real performance ($\mu = 0.25$ for $pfmc_{real}$) and a low confidence ($\mu = 0.25$ for *cnfd*): (a) real performance; (b) observed performance; (c) misjudgement; (d) confidence.

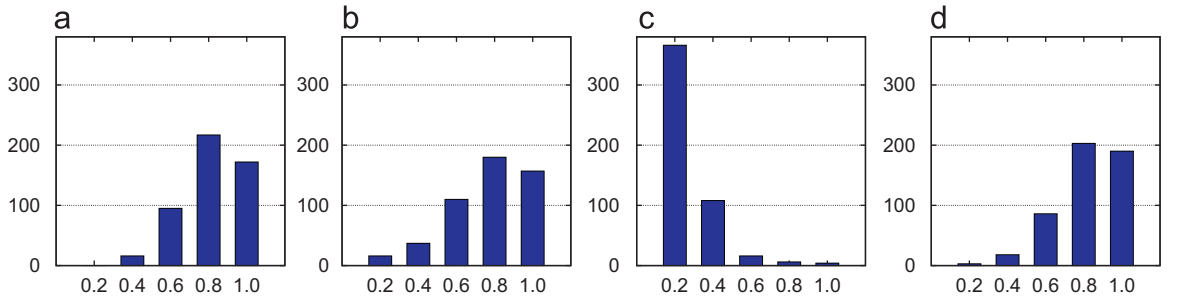


Fig. 6. Generated worker population according to a high real performance ($\mu = 0.75$ for $pfmc_{real}$) and a high confidence ($\mu = 0.75$ for *cnfd*): (a) real performance; (b) observed performance; (c) misjudgement; (d) confidence.

histograms of the workers' main simulation properties with low real performance ($\mu = 0.25$ for $pfmc_{real}$) and a low confidence ($\mu = 0.25$ for *cnfd*). With this configuration, the average real performance, i.e., the average level of all skills, of most workers is between 0.2 and 0.4 (cf., Fig. 5(a)), followed by those between 0 and 0.2. The observed performance displayed in Fig. 5(b) for most worker is also low. Fig. 5(c) indicates that a majority (40%) of the workers has a difference between 0.2 and 0.4 for real performance and observed performance. The confidence in most workers' skills is relatively low (cf., Fig. 5(c)). To sum up, this is a scenario with many unexperienced workers and a crowdsourcing system that does not have a profound knowledge about the workers.

Fig. 6 displays the same information about workers for a system with high real performance ($\mu = 0.75$ for $pfmc_{real}$) and a high confidence ($\mu = 0.75$ for *cnfd*). In this scenario the average real performance is much higher and the observed performance is much closer to the real one. The misjudgement error for the majority of all workers is below 0.2, i.e., their true skills can be estimated much more correctly. This situation is also reflected in the confidence. This scenario refers to a well-established crowdsourcing system with highly skilled workers.

Tasks. All tasks are generated randomly with the following methodology. The real hidden result of a task is set to a uniformly distributed random value $\mathcal{U}(0,1)$ from the range [0,1]. An expected duration is randomly drawn from the set $\{1,2,\ldots,24\}$. For each of the four skills a fair coin is tossed determining whether the task requires the respective skill. After that, weights are randomly assigned to each skill determining the importance of the respective skill for the task. The sum of those weights always equals to 1. Last but not least, a random deadline is generated for which is guaranteed that it is after the expected duration.

Requester behavior. In every round of the simulation each requester is asked to submit a task. After processing, requesters receive the result for the task. If they receive the result after the deadline they rate the transaction with a value of zero and suspend for 20 rounds, i.e., they would refuse to issue a new task due to the negative experience. If the result is transmitted on time requesters rate the quality of the received result. Computationally this is done by comparing the task's real result with the received result. It is assumed that task requesters are able to estimate whether the received result is close to what was expected. The best possible rating is one, zero is the worst result, all values in between are possible. Ratings for a worker w and task t, be it negative or positive, increase the confidence $cnfd(w,s)$ and update the observed performance $pfmc(w,s)$ of w's skills, weighted by the weight of skill s in task t. This means that a rating for a task requiring mostly skill s_1 and a little bit of skill s_2 will mainly modify the confidence and performance values for s_1 (i.e., $pfmc(w,s_1)$ and $cnfd(w,s_1)$) and only slightly influence the values for s_2 (i.e., $pfmc(w,s_2)$ and $cnfd(w,s_2)$). If the rating is below a threshold of 0.5 the worker suspends for 10 rounds, similar to a deadline violation. Hence, requesters with negative experiences tend to make less usage of the crowdsourcing marketplace. In addition to the pure task description requesters announce the maximum price they are willing to pay for the processing of the task. Prices are also represented by random values within the range [0,1]. Tasks with high quality and high

Please cite this article as: B. Satzger, et al., Auction-based crowdsourcing supporting skill management, Information Systems (2012), http://dx.doi.org/10.1016/j.is.2012.09.003

expected duration are more likely to have costs close to the maximum value.

Worker behavior. When asked for a bid during an auction a worker first checks whether it is realistic to finish the task before the deadline considering all tasks the worker is working on. Each task has an expected duration d and each worker would only submit a bid if s/he has at least $1.3 \cdot d$ of time to work on the task before the end of the deadline considering already accepted tasks. The actual processing time for a task t is set to a random value $d+|\mathcal{N}(0,\sigma^2)|$, where the variance is higher for workers with low real performance, regarding skills needed for processing t. Hence, for workers with high real performance the processing time is more likely to be close to t. This value is set by the simulation environment and the workers do not know about the exact processing time in advance. Workers determine the price according to a linear combination of a task's effort (i.e., a normalized value of the expected duration) and her/his workload. The rationale is that workers want more money for work-intensive tasks; the higher a worker's current workload the more payment is conceived:

$$bid(w,t) = 0.5 \cdot effort(t) + 0.5 \cdot load(w).$$

The processing of a task has a positive influence on the worker w's real performance of all skills involved, i.e., $pfmc(w,s)_{real,new} = pfmc_{real}(w,s) + 0.1 \cdot (1 - pfmc_{real}(w,s)) \cdot weight(s,t)$. This modeling of a training effect results in a high learning rate for workers with low real performance and a slowed down learning effect for workers who are already very good. The term $weight(s,t)$ refers to the weight of skill s in task t, i.e., how important skill s is for processing task t. Hence, more important skills are trained more strongly than skills that are less needed for processing a task.

Auction processing. Auctions are conducted for the purpose of matching a task to a worker. As described in Section 4 there is a qualification and preselection stage before the actual auction in order to avoid spamming a huge worker base with auction request for which many workers may not have the necessary skills. Since we only consider a limited number of 500 workers it is reasonable to admit all of them to the auctions. To achieve that the aptitude function, see Eq. (1), is set as follows:

$$aptitude : (w,t) \mapsto 1.$$

After receiving the workers' bids they are ranked by a ranking function as defined in Eq. (2):

$$rank : (w,t,b) \mapsto 0.7 \cdot suitability(w,t) + 0.3 \cdot (1 - price(b)).$$

The function suitability outputs how suitable a worker w is to process task t:

$$suitability : (w,t) \mapsto \sum_{s \in S} pfmc(w,s) \cdot cnfd(w,s) \cdot weight(s,t).$$

Workers may either return a bid or refuse to submit a bid. From the received bids all values are removed whose price is higher than the price the requester is willing to pay. The remaining valid bids are ranked such that a high observed performance, high confidence, and a low price of the bid positively influence the rank. The emphasis at that stage clearly is on the performance and not on the price. It may happen that there is no valid bid; in that case the requester is informed that the task could not be processed.

Skill evolution. In this work we want to investigate how crowdsourcing can benefit from skill evolution, which is achieved by assigning assessment tasks to workers. This is especially useful for workers with a low average confidence value. For these workers only few or no ratings are available. An assessment task is a task that is assigned to a worker although another worker has won the auction and was assigned to the task as well. The workers are not aware of the fact that there are other workers processing the very same task; requesters are not either. The crowdsourcing provider is responsible for paying for the training tasks. As usual, the result of the highest ranked worker is returned to the requester but it is additionally used as a reference for the training task. This enables the marketplace to generate a rating for the assessed worker by comparing her/his result to the reference. A further positive effect is the training of the assessed worker. The assignment of training tasks is based on the received list of valid bids. For controlling the skill evolution equation (3) needs to be set accordingly.

The following definition of the assessment function, which results in disabling skill evolution and leads to purely profit driven auction decisions, maps each combination of workers, bids, and reference workers to 0:

$$assess_{profit} : (w,b,w_r) \mapsto 0.$$

In the evaluation we have used the following setting for the skill evolution enabled auctions:

$$assess_{skill} : (w,b,w_r) \mapsto \begin{cases} 0 & \text{if working queue not empty} \\ & \text{or } suitability(w_r,t) < 0.9, \\ select(w,b) & \text{otherwise.} \end{cases}$$

The function $assess_{skill}$ guarantees that only workers with empty working queue are assessed and that reference workers are a very good match for the task. This is crucial because the worker w is rated according to the result of the reference worker w_r. If workers with a suitability less than 0.9 win an auction a training task assignment is prohibited. If all prerequisites are met the *select* function determines the workers who are assigned a training task. It is possible that multiple training tasks are assigned.

$$select : (w,b) \mapsto \begin{cases} 1 & \text{with probability } benefit(w,t), \\ 0 & \text{otherwise.} \end{cases}$$

The *select* function assigns a training task based on the function benefit which outputs how much the system benefits from assigning training task t to worker w. A high benefit increases the likelihood for a training task. The benefit function is defined as follows:

$$benefit : (w,t) \mapsto \sum_{s \in S} urg(s) \cdot weight(s,t) \cdot pfmc(w,s) \cdot (1 - cnfd(w,s)).$$

The above definition increases benefit if the urgency $urg(s)$ for the skill is high, the observed performance of the worker for the skill is high, but the confidence of the system in the worker is still low. The urgency is

Please cite this article as: B. Satzger, et al., Auction-based crowdsourcing supporting skill management, Information Systems (2012), http://dx.doi.org/10.1016/j.is.2012.09.003

determined by the crowdsourcing system; if a skill $s \in S$ is rare then the urgency $urg(s)$ is close to 1, and close to 0 if there is plenty of workers with that skill.

5.3. Results

In the following we present the outcomes of our experiments. In order to avoid major variations in the result values each experiment scenario has been run 50 times. Hence, the actual values given in Tables 1–3 are averaged results. Each row of these tables represents one of the scenarios as specified in Fig. 4. For each experiment we recorded the ratings issued by the requester, the misjudgement (*Misj.*), i.e., the gap between real and observed performance, the total number of tasks issued (*Issued*), the total number of tasks returned after the deadline (*Late*), the number of tasks with empty bids, i.e., that have received no bid (*Empty*), and the number of conducted training/assessment tasks (*Train.*).

Table 1
Result table with five requesters (low load).

Scenario		Results					
$pfmc_{real}$	*cnfd*	Rating	Misj.	Issued	Late	Empty	Train.
$\mu=0.25$	$\mu=0.25$	0.579	0.292	970	117	3	–
		0.568	0.285	934	121	3	129
$\mu=0.25$	$\mu=0.5$	0.652	0.233	1256	100	3	–
		0.658	0.219	1288	101	2	425
$\mu=0.25$	$\mu=0.75$	0.711	0.146	1627	83	2	–
		0.713	0.138	1648	87	3	730
$\mu=0.5$	$\mu=0.25$	0.803	0.310	2423	67	3	–
		0.817	0.198	2562	66	2	2779
$\mu=0.5$	$\mu=0.5$	0.846	0.249	3056	51	2	–
		0.857	0.144	3245	47	2	3908
$\mu=0.5$	$\mu=0.75$	0.884	0.159	3794	33	3	–
		0.894	0.099	3909	34	3	4593
$\mu=0.75$	$\mu=0.25$	0.932	0.300	4295	25	3	–
		0.941	0.094	4446	20	3	8041
$\mu=0.75$	$\mu=0.5$	0.949	0.238	4584	16	2	–
		0.954	0.076	4633	16	3	8699
$\mu=0.75$	$\mu=0.75$	0.966	0.154	4791	10	2	–
		0.970	0.065	4792	10	2	8561

Table 2
Result table with 50 requesters (medium load).

Scenario		Results					
$pfmc_{real}$	*cnfd*	Rating	Misj.	Issued	Late	Empty	Train.
$\mu=0.25$	$\mu=0.25$	0.520	0.218	7573	1709	247	–
		0.524	0.165	7623	1678	252	1965
$\mu=0.25$	$\mu=0.5$	0.548	0.173	8209	1634	266	–
		0.549	0.143	8182	1617	308	1772
$\mu=0.25$	$\mu=0.75$	0.581	0.123	9011	1571	239	–
		0.578	0.112	8935	1577	308	1693
$\mu=0.5$	$\mu=0.25$	0.759	0.206	16 391	1298	299	–
		0.780	0.093	18 119	1236	475	6009
$\mu=0.5$	$\mu=0.5$	0.783	0.176	18 334	1223	386	–
		0.793	0.089	19 470	1200	594	6105
$\mu=0.5$	$\mu=0.75$	0.798	0.129	20 004	1168	475	–
		0.804	0.083	20 682	1156	826	5983
$\mu=0.75$	$\mu=0.25$	0.911	0.207	32 375	810	203	–
		0.928	0.046	34 650	722	382	13 840
$\mu=0.75$	$\mu=0.5$	0.919	0.174	33 966	739	296	–
		0.932	0.047	35 651	681	480	14 155
$\mu=0.75$	$\mu=0.75$	0.928	0.124	35 867	667	283	–
		0.936	0.046	36 425	662	408	14 349

Table 3
Result table with 100 requester (high load).

Scenario		Results					
$pfmc_{real}$	*cnfd*	Rating	Misj.	Issued	Late	Empty	Train.
$\mu=0.25$	$\mu=0.25$	0.484	0.169	15 038	3666	2288	–
		0.498	0.127	15 588	3621	2512	2095
$\mu=0.25$	$\mu=0.5$	0.496	0.144	15 523	3648	2355	–
		0.508	0.118	16 024	3579	2550	1750
$\mu=0.25$	$\mu=0.75$	0.515	0.112	16 211	3586	2372	–
		0.530	0.101	17 005	3538	2848	1593
$\mu=0.5$	$\mu=0.25$	0.732	0.130	29 442	2960	3251	–
		0.746	0.087	32 045	2855	4563	4029
$\mu=0.5$	$\mu=0.5$	0.742	0.119	31 100	2860	3691	–
		0.753	0.084	33 316	2784	4922	3924
$\mu=0.5$	$\mu=0.75$	0.757	0.099	33 166	2818	4130	–
		0.767	0.079	35 475	2669	5223	3882
$\mu=0.75$	$\mu=0.25$	0.893	0.111	55 344	2111	3486	–
		0.902	0.046	57 539	2048	4315	8551
$\mu=0.75$	$\mu=0.5$	0.896	0.103	56 623	2051	3511	–
		0.907	0.046	59 806	1946	4523	8701
$\mu=0.75$	$\mu=0.75$	0.902	0.082	58 917	1966	3974	–
		0.908	0.048	60 399	1924	4677	8540

Table 1 represents the results of scenarios including five requesters. The results show that there is no overload situation in any scenarios, because *Empty* remains between 2 and 3 tasks, which means that almost always at least one suitable bid is received for a task. This suggests that there is enough workers in the environment for handling the few tasks. Therefore, training only slightly improves the rating; in some cases training even marginally deteriorates the ratings. However, training helps to decrease the misjudgement in all cases. The first three rows represent scenarios where the average skill level is very low, which manifests in low ratings given by the requesters and only few issued tasks. If requesters have to give bad ratings they are unsatisfied and issue fewer tasks. A higher confidence, i.e., better knowledge about the workers due to more available ratings, significantly improve the ratings, because the mapping is based on a more solid fundament. Interestingly, for the case of very low average real performance, training tasks slightly decrease the rating and causes more late submissions. This is because the few high-potentials are additionally occupied with assessment tasks; therefore real tasks are either late or processed by less suitable workers, resulting in decreased quality and worse ratings. In a scenario with many skilled people but few ratings (e.g., $pfmc_{real}=0.75$, $cnfd=0.25$) the training leads to significant improvements in terms of rating and misjudgement. The last three rows represent platforms with workers of relatively high skill levels and different amounts of rating data. Assessment tasks lead to a improvement in the ratings, which represents how satisfied the requesters are. For lower levels of confidence the gains of assessment tasks are greater than for settings where the system already knows much about their workers. The number of late task completion

Please cite this article as: B. Satzger, et al., Auction-based crowdsourcing supporting skill management, Information Systems (2012), http://dx.doi.org/10.1016/j.is.2012.09.003

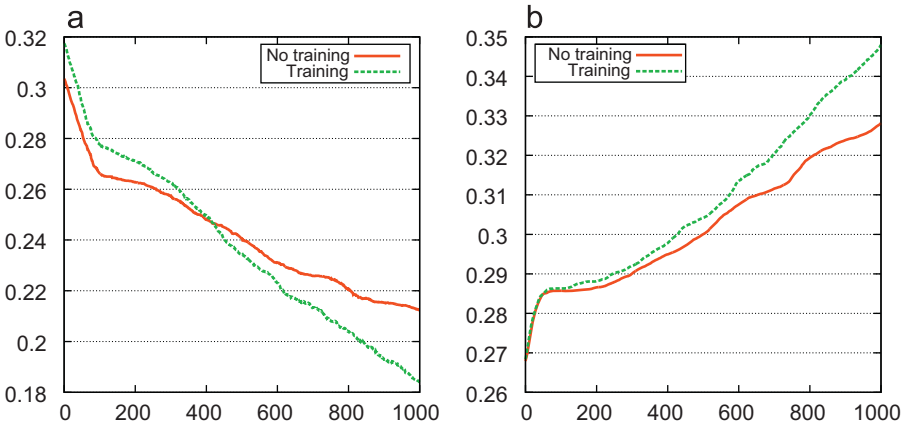


Fig. 7. Development of misjudgement and real performance over time with and without skill evolution for a single experiment of the scenario with low real performance ($\mu = 0.25$ for $pfmc_{real}$) and low confidence ($\mu = 0.25$ for $cnfd$). The x-axis represents the time of the experiment and the y-axis shows the respective misjudgement and real performance values: (a) misjudgement; (b) real performance.

decreases with both better workers and more knowledge about the workers. The number of tasks issued and the rating directly correlates.

The situation is somewhat different in Table 2 with 50 requesters, i.e., a 10 times higher load. With more tasks issued, training becomes more important since workers are more rare and additional skilled workers help to cope with the amount of tasks. Hence, with training the ratings are always better. Assessment tasks, of course, always help to reduce the misjudgement because of additional ratings for promising workers with low number of ratings. However, assessment tasks also put additional load onto the system and although this is tried to be minimized by giving assessment tasks only to workers who are not busy, it still involves a tradeoff. It materializes with a higher number of auctions where no bid is submitted, which can be attributed partially to the additional load caused by assessment tasks. Another reason is that requesters issue more tasks as they are more satisfied. Compared to the platform with low load (Table 1), the respective ratings for the platform with 10 times higher load are always worse. The reason is that also weaker workers tend to win auctions if high-performers are busy and cannot bid at all or submit bids with too high prices. Busy workers tend to request more reward in their bids than idle workers. If a worker has no free capacities s/he does not submit a bid to an auction, even if invited.

Finally, Table 3 illustrates the results of scenarios with 100 requesters. With twice as many requesters as in the previous scenario there is around twice as many late tasks and 10 times as many empty tasks, which refers to an overload situation. Even under such circumstances training tasks pay off with relation to the satisfaction of the requesters and the number of tasks issued. There is more empty tasks when assessment tasks are used but the overall throughput of tasks is still higher because overall requesters issue more tasks. Compared to the settings with low and medium loads, the ratings in the high load setting are generally worse. The number of assessment tasks is lower than with 50 requesters, even though there are more possible opportunities for assessment tasks. This is because we do not give assessment tasks to busy workers.

Runtime results. The results presented in the tables represent final averaged results after a simulation period of 1000 steps, i.e., each single scenario consists of 1000 time units and is also rerun 100 times to ensure meaningful results. In the following we give a better insight into the processes of single experiments during runtime. In Figs. 7 and 8 we present the temporal change for the values of misjudgement and real performance. The x-axis represents the time measured in time units. Both figures show the results of a simulation with 50 requesters, i.e., a setting that refers to medium load, as presented in Table 2. Fig. 7(a) details the decrease of the misjudgement for a setup with an average real performance of $\mu = 0.25$ ($pfmc_{real}$), and average confidence of $\mu = 0.25$ ($cnfd$). This refers to a setting with few skilled workers and few ratings. As during the run of the experiment more and more ratings are available, the crowdsourcing platform is able to better estimate the workers' skill levels and the misjudgement factor decreases. Whilst the decline of both curves has an almost similar progress for the first 400 rounds, afterwards, training assisted worker assessment decreases the gap of misjudgement more rapidly than without training. Initially requesters offer a task each round, however due to many low quality processing the requesters become unsatisfied and submit less tasks. This explains the kink at around 75 time units. The fact that the initial misjudgement is higher for the case with training can be attributed to random effects in the single run of the experiment that is shown. Fig. 7(b) shows the improvement of the real performance due to training effects. We assume that the workers slightly improve all skills relevant for a task when they process it. With assessment tasks, the overall average performance of the workers increases more rapidly because altogether they process more tasks.

Fig. 8 shows the same graphs for a platform with the same load as above, but involving highly skilled workers and already many available ratings. In such a scenario assessment tasks work very well because workers with

Please cite this article as: B. Satzger, et al., Auction-based crowdsourcing supporting skill management, Information Systems (2012), http://dx.doi.org/10.1016/j.is.2012.09.003

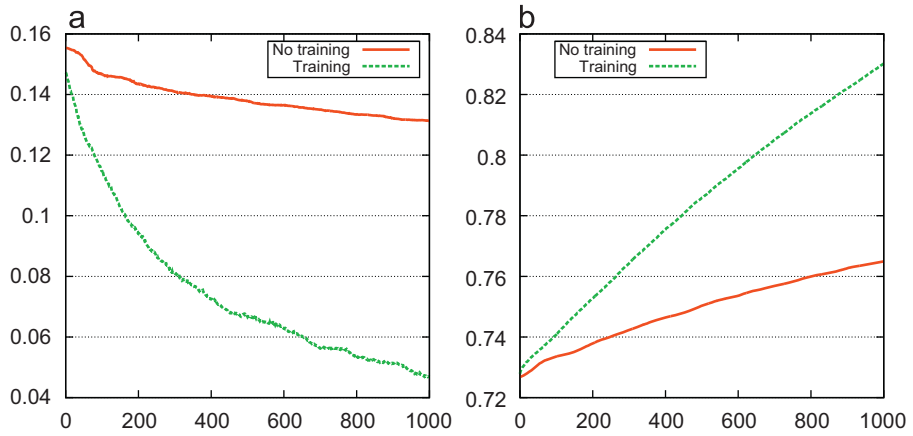


Fig. 8. Development of misjudgement and real performance over time with and without skill evolution for a single experiment of the scenario with high real performance ($\mu = 0.75$ for $pfmc_{real}$) and high confidence ($\mu = 0.75$ for $cnfd$). The x-axis represents the time of the experiment and the y-axis shows the respective misjudgement and real performance values: (a) misjudgement; (b) real performance.

few or no ratings can be assessed by comparing them with many competent reference workers. Fig. 8(a) shows how quickly unknown workers can be integrated and judged very accurately with assessment tasks compared to a situation without training. Also, skill evolution helps to increase performance significantly, because in this scenario with many skilled workers, newcomers have hardly the opportunity to develop their skills without assessment tasks. This problem is mitigated with the use of assessment tasks, as shown in Fig. 8(b), i.e., the overall performance increases much more quickly.

5.4. Discussion of results

The utilization of auctions for crowdsourcing of tasks has the advantage that the requesters do not have to prespecify the exact price, which may be too high or too low. For controlling their expenses they can define a maximum price. In addition to this convenient property, the conducted experiments show that auctions actually work in realistic settings. In a previous work [9] we investigated two simple scenarios; in both scenarios assessment tasks were able to improve the crowdsourcers' satisfaction. Due to the much more detailed experiments conducted here we can derive more sophisticated conclusions and reason about the effects of tasks requiring multiple skills. Assessment tasks improve the results if either the worker base has a good average skill level or the confidence in the workers' skills is high. The latter is the case when enough ratings have been provided. If the average skills of the crowd is high we can identify many high-performers using assessment task and, thus, can quickly strengthen the workforce. In cases where the confidence in the workers is high it is very likely to at least choose the right workers as reference for automatic tandem task assessment and to correctly identify high-performers. When the platform load is low, the crowd has low average skills, and only few ratings are available, then assessment tasks can lead to decreased performance. This effect is reinforced when considering multiple skills as shown by comparing results with [9]. With multiple skills each worker is much more unique compared to "narrow" platforms, for which all workers have the same skill set. When multiple diverse skills are required it is much harder to find suitable reference workers. Choosing unsuitable reference workers means that the ratings that are generated by assessment tasks are unreliable and that additional load is put onto the few capable workers for little gain.

6. Conclusions and future work

For crowdsourcing of tasks in an enterprise setting and seamless integration into business workflows it is essential to a have a high degree of automation and quality assurance. In this paper we propose a crowdsourcing marketplace based on auctions. We extend classical sealed-bid auctions often used in procurement to ensure high quality; only suitable workers are invited and bids are ranked not only according to the specified reward but also take the users' profiles into account. Rating information provided by requesters are the most important ingredient we use for generating user profiles. Compared to crowdsourcing platforms in which users have to search through all tasks to find interesting ones in our approach workers are only confronted with tasks they are suitable for. Crowdsourcers do not have to explicitly specify the reward for a task but the prices are generated by supply and demand. A further extension to the auctioning mechanisms allows for automated creation of assessment tasks that aim at helping new workers to ramp up and the platform to generate more accurate profiles. We conduct extensive experiments for crowdsourcing of tasks that require multiple different skills and investigate the effects of assessment tasks on different quality metrics. We discuss the results for 54 different scenarios and come to the conclusion that such a skill evolution mechanisms pays off in most settings; generally, they give workers the chance to prove their skills and allow the crowdsourcing system to better estimate its members, which helps to make more informed decisions.

Please cite this article as: B. Satzger, et al., Auction-based crowdsourcing supporting skill management, Information Systems (2012), http://dx.doi.org/10.1016/j.is.2012.09.003

As part of our ongoing research we plan to investigate how to incorporate tasks requiring collaboration. Workers may, for instance, decompose tasks into subtasks, "crowdsource" the subtasks, and finally assemble the partial results into the final one. In such a setting the crowd would contribute the knowledge how to compose and assemble complex tasks. Furthermore, crowd members could provide higher level services such as quality control and insurance. Apart from auction-based mechanisms, specifications such as WS-HumanTask [12] and BPEL4People [8] for modeling human interactions in service-oriented business environments need to be extended to cope with the dynamics inherent to open crowdsourcing platforms. For example, providing skill and quality models based on prior negotiated service-level agreements (SLAs) that augment WS-HumanTask's people assignment model.

Acknowledgment

This work received funding from the EU FP7 program under the agreements 215483 (SCube) and 257483 (Indenica).

References

[1] D. Brabham, 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.
[2] M. Vukovic, Crowdsourcing for enterprises, in: Proceedings of the 2009 Congress on Services, IEEE Computer Society, 2009, pp. 686–692.
[3] A. Doan, R. Ramakrishnan, A.Y. Halevy, Crowdsourcing systems on the world-wide web, Communications of the ACM 54 (4) (2011) 86–96.
[4] Amazon Mechnical Turk 〈http://www.mturk.com〉 (last access Sep. 2012).
[5] P.G. Ipeirotis, Analyzing the amazon mechanical turk marketplace, SSRN eLibrary 17 (2) (2010) 16–21.
[6] CrowdFlower 〈http://crowdflower.com/〉 (last access Sep. 2012).
[7] D. Schall, H.-L. Truong, S. Dustdar, Unifying human and software services in web-scale collaborations, IEEE Internet Computing 12 (3) (2008) 62–68.
[8] M. Kloppmann, D. Koenig, F. Leymann, G. Pfau, A. Rickayzen, C. von Riegen, P. Schmidt, I. Trickovic, Ws-bpel extension for people–bpel4people, Joint white paper, IBM and SAP, 2005.
[9] B. Satzger, H. Psaier, D. Schall, S. Dustdar, Stimulating skill evolution in market-based crowdsourcing, in: 9th International Conference on Business Process Management (BPM '11), Lecture Notes in Computer Science, Springer, 2011, pp. 66–82.
[10] D. Schall, B. Satzger, H. Psaier, Crowdsourcing Tasks to Social Networks in BPEL4People, World Wide Web, Springer, 2012.
[11] R. Khazankin, B. Satzger, S. Dustdar, Optimized execution of business processes on crowdsourcing platforms, in: 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom '12), IEEE, 2012.
[12] A. Agrawal, M. Amend, M. Das, M. Ford, C. Keller, M. Kloppmann, D. König, F. Leymann, R. Müller, G. Pfau, et al., Web services human task (ws-humantask), version 1.0, available at 〈http://incubator.apache.org/hise/WS-HumanTask_v1.pdf〉, 2007.
[13] F. Leymann, Workflow-based coordination and cooperation in a service world, in: Cooperative Information Systems (CoopIS '06), Lecture Notes in Computer Science, Springer, 2006, pp. 2–16.
[14] C. Castellano, S. Fortunato, V. Loreto, Statistical physics of social dynamics, Reviews of Modern Physics 81 (2) (2009) 591–646.
[15] D. Brabham, Crowdsourcing: a model for leveraging online communities, The Routledge Handbook of Participatory Culture, 2012.
[16] R. Kumar, Y. Lifshits, A. Tomkins, Evolution of two-sided markets, in: Proceedings of the 3rd ACM International Conference on Web Search and Data Mining (WSDM '10), ACM, New York, NY, USA, 2010, pp. 311–320.
[17] Y.A.S. System, Points and levels, http://answers.yahoo.com/info/scoring_system (last access Sep. 2012).
[18] Q. Su, D. Pavlov, J.-H. Chow, W. C. Baker, Internet-scale collection of human-reviewed data, in: Proceedings of the 16th International Conference on World Wide Web, WWW '07, ACM, New York, NY, USA, 2007, pp. 231–240.
[19] J. Ross, L. Irani, M. Silberman, A. Zaldivar, B. Tomlinson, Who are the crowdworkers?: shifting demographics in mechanical turk, in: 28th International Conference on Human Factors in Computing Systems (CHI '10), ACM, 2010, pp. 2863–2872.
[20] L.B. Chilton, J.J. Horton, R.C. Miller, S. Azenkot, Task search in a human computation market, in: SIGKDD Workshop on Human Computation (HCOMP '10), ACM, 2010, pp. 1–9.
[21] P.G. Ipeirotis, F. Provost, J. Wang, Quality management on amazon mechanical turk, in: SIGKDD Workshop on Human Computation (HCOMP '10), ACM, 2010, pp. 64–67.
[22] P. Wais, S. Lingamneni, D. Cook, J. Fennell, B. Goldenberg, D. Lubarov, D. Martin, Towards building a high-quality workforce with mechanical turk, in: Workshop on Computational Social Science and the Wisdom of Crowds (held at NIPS '10), 2010.
[23] Crowd Control 〈http://www.crowdcontrolsoftware.com/〉 (last access Sep. 2012).
[24] Samasource 〈http://samasource.org/〉 (last access Sep. 2012).
[25] H. Psaier, L. Juszczyk, F. Skopik, D. Schall, S. Dustdar, Runtime behavior monitoring and self-adaptation in service-oriented systems, in: 4th International Conference on Self-Adaptive and Self-Organizing Systems (SASO '10), IEEE, 2010, pp. 164–173.
[26] P. Klemperer, Auctions: Theory and Practice, Toulouse Lectures in Economics, Princeton University Press, 2004.
[27] D. DiPalantino, M. Vojnovic, Crowdsourcing and all-pay auctions, in: Proceedings of the 10th ACM Conference on Electronic Commerce (EC '09), ACM, New York, NY, USA, 2009, pp. 119–128.
[28] T.X. Liu, J. Yang, L.A. Adamic, Y. Chen, Crowdsourcing with all-pay auctions: a field experiment on taskcn, Proceedings of the American Society for Information Science and Technology 48 (1) (2011) 1–4.
[29] S. Chawla, J.D. Hartline, B. Sivan, Optimal crowdsourcing contests, in: Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '12), SIAM, 2012, pp. 856–868.
[30] U. Gneezy, R. Smorodinsky, All-pay auctions—an experimental study, Journal of Economic Behavior & Organization 61 (2) (2006) 255–275.
[31] P. Milgrom, Auctions and bidding: a primer, The Journal of Economic Perspectives 3 (3) (1989) 3–22.
[32] G. Paolacci, J. Chandler, P. Ipeirotis, Running experiments on amazon mechanical turk, Judgment and Decision Making 5 (5) (2010) 411–419.

Please cite this article as: B. Satzger, et al., Auction-based crowdsourcing supporting skill management, Information Systems (2012), http://dx.doi.org/10.1016/j.is.2012.09.003