Improving wiki article quality through crowd coordination: a resource allocation approach

ABSTRACT

In this paper we propose a crowd coordination mechanism to increase the quality of articles produced in wiki systems. Wikis constitute promising social digital ecosystems for collaborative knowledge creation on the Web. However, as a result of the purely self-coordinated manner that they function, they cannot ensure the quality of the produced articles - an issue that affects their reliability and acceptance. We show that wiki article quality optimization can be formulated as a resource allocation problem. Contributors are selected from the wiki community crowd according to their skills, and matched to the articles they can improve the most. A model of the English Wikipedia is given, parameterized and validated from recent field studies results. Experimental results, obtained with simulation systems implementing this model, indicate that the proposed mechanism can lead to the production of wiki articles of higher quality, compared to the respective results achieved by the fully self-coordinated wiki.

Keywords: intelligent system; wikis; collaborative content creation; crowd coordination; resource allocation.

INTRODUCTION

Wikis are among the most popular technologies for collaborative knowledge production. In brief, a wiki is a collaborative content creation system, where users contribute knowledge content in the form of articles, while they can also edit and even delete the contributions of others (Louridas, 2006). Wikis have received significant interest in the past few years and they are increasingly being used to support knowledge development in many domains, such as education, research and scientific collaboration, activities of the public sector and in corporate environments. Last, one of the most popular wikis, Wikipedia, is an ever-growing source of information and a large social digital ecosystem with millions active users and articles.

The rapid expansion and success of wikis is based on the open form of user collaboration that they are based on. That is, wiki users are free to edit any article they wish, with almost no restrictions on their access and edit rights. This open collaboration enables the massive production of wiki articles, which cover broad spectrum of topics and expertise backgrounds. However, this same self-coordination poses significant limitations, in terms of content qualityand timeliness (Denning, Horning, Parnas, & Weinstein, 2005). Take as an example Wikipedia: although a number of qualitative articles with many user contributions may be found, the majority of articles are still of low quality with only a few contributions (Lam & Riedl, 2009a).

This inability to guarantee quality lowers the reliability of wikis and hinders their adoption (Liu & Ram, 2011).

Recent research suggests that a solution to the above quality problem is to reinforce the self-coordination pattern of wikis with more formal algorithm-based coordination schemas that will guide the wiki crowd, systematize contributions and help them utilize their knowledge competencies more efficiently (Kittur, Lee, & Kraut, 2009; O'Mahony & Ferraro, 2007). Despite the above, research efforts towards this direction are still very few.

In this paper we argue that the development of such a coordination schema can be viewed as a resource allocation problem. The wiki is seen as a system, with resources, which are the users and their expertise, and tasks, which are the wiki articles that need quality improvement. The allocation objective is to match users to articles, in such a way as to maximize the average quality of the articles inside the wiki. The system also needs to respect specific constraints, such as the maximum individual user workload.

To solve the above, we propose an allocation-based coordination mechanism. The mechanism monitors the system's availability in resources (users), and tasks (wiki articles below a certain quality threshold). It then calculates which user should contribute to which article, in order to maximize overall wiki quality, and recommends the article to the user. The allocation of users to articles is performed through a heuristic greedy algorithm.

The contributions of this work are therefore the following:

- Highlight the need for system-level coordination to improve the performance of current mass crowd participation systems, such as wikis.
- Formulate the problem wiki crowd coordination as a resource allocation problem.
- Propose and examine an allocation algorithm tailored to the above problem.

The rest of this paper is organized as follows: Section 2 presents the related literature and positions this work in respect to relevant research efforts. Section 3 presents the proposed wiki coordination mechanism and formulates the resource allocation problem. Section 4 presents and analyzes the obtained experimental results. Section 5 discusses the open issues and perspectives. Finally, section 6 presents future research directions and concludes the paper.

RELATED LITERATURE

Quality amelioration mechanisms in wikis

The main quality improvement mechanism of wikis is casual users themselves, who read the articles, evaluate and gradually improve them. The problem with this quality improvement mechanism is that it is ad-hoc, and often coincidental: in case a user happens to read an article that he is knowledgeable about, he may improve it. But very often the information about which article needs improvement is not communicated to knowledgeable users (at best it is left in the article's discussion page), thus resulting to a very long tail of articles with few contributions and a low quality (Lam & Riedl, 2009b).

To amend this, some wikis reinforce their quality mechanisms using "wiki gardeners". These are people, usually among the most expert wiki users, who manually check for low quality content and either improve it themselves, or identify the users who should be asked to contribute

(Welser, et al.). The limitation of this solution however is that it cannot work at large-scale, since the experts can inevitable process only a small number of articles (Butler, Joyce, & Pike, 2008).

Bots are a third quality mechanism. Bots are programs running inside the wiki, which automatically perform content maintenance tasks such as link repairing, template text expansion and vandalism reversion. The advantage of bots is that they can process many more articles compared to humans. Their limitation is that typically bots are designed for simple tasks and not for identifying which article needs improvement or which user should be asked to improve it.

In line with the above, recent literature indicates that in order to achieve high quality for all articles inside the wiki we need algorithmic mechanisms, which will coordinate the wiki crowd and systematize their contributions (Barberio & Lomi, 2009; Kittur, et al., 2009; O'Mahony & Ferraro, 2007). The advantage in using AI methods for coordinating user contributions is that AI can shift the extra load of quality control and expert identification from human users, who are therefore free to focus only on knowledge contribution activities. Despite the above, almost no works to-date propose a wiki-wide coordination mechanism. Relative works however of this kind have started to emerge in the neighboring discipline of crowdsourcing.

Crowd coordination mechanisms

Most works on algorithm-based crowd coordination focus on crowdsourcing markets, such as Amazon's Mechanical Turk[1]. Research in this domain aims mostly at coordinating the contributions of users regarding simple tasks (like captcha recognition), in order to minimize the total paid cost. Similarly to the present paper, many of these works employ resource allocation (Psaier, Skopik, Schall, & Dustdar, 2011) to coordinate the contributions of people, while other methods such as queue theory (Bernstein, Karger, Miller, & Brandt., 2012), mechanism design (Nath, Zoeter, Narahari, & Dance, 2011) and game theory (Ghosh & Hummel, 2012) can also be found. Their very successful results indicate that AI-based coordination methods can indeed be used to optimize the performance of a user crowd. However they cannot be applied as such on the wiki quality problem because of two main differences with the latter: first it involves more complex tasks, i.e. knowledge creation, rather than the contribution to simple micro-tasks and secondly here the optimization goal is quality rather than cost.

Finally, for the domain of wikis, hardly any crowd coordination works exist to date with a focus on optimizing for quality. The most relevant studies applying AI in wikis, use recommendation approaches proposing to users articles matching most with their interests (Hoffmann, et al., 2009; Kong, Hanrahan, Weksteen, Convertino, & Chi, 2011). These approaches have two main differences with the approach presented in this paper. First they focus on improving user participation and not article quality. Second, they work on a user and not system-level optimization goal. This means that their article recommendations aim at increasing the participation of each specific individual, and not at achieving an improvement of the global article quality. However by using user-level recommendations, the enhancement of quality of articles is not controlled or balanced. Indeed, it may well be the case that an article receives many contributions and therefore reaches high quality levels, while other articles are never suggested and therefore remain at very low quality levels. This problem is also well-known in the area of traditional recommender systems, which typically function at user-level, and it is referred to as “diversification” problem (Yu, Lakshmanan, & Amer-Yahia, 2009). It becomes

[1] https://www.mturk.com

therefore apparent that in order to ensure a minimum of quality for all articles inside the wiki, working with user-level article recommendations is not enough. Instead a solution is needed that works at system-level and suggests articles to users taking into account not only their individual preferences but also a global system objective, i.e. the maximization of article quality across the wiki.

In this paper we propose a coordination mechanism for wikis, which is designed to optimize article quality at system-level. The proposed mechanism monitors the overall quality needs of the wiki (which article need to be enhanced and how much), the availability of the wiki in experts and then, with this "broader picture" in mind, it suggests articles to users, so that the overall average article quality will be maximized. The optimization of the global quality target is viewed as an allocation problem and handled by an allocation algorithm. The individual user factor is also taken into account, since the suggestions of the mechanism are partially driven by the expertise, and therefore the interest, of users to the different knowledge topics covered by the wiki.

METHODOLOGY

Wiki crowd coordination mechanism: Overall functionality

The overall wiki coordination mechanism functions as follows. For every new or modified article in the wiki, the mechanism evaluates the article's quality, as a single numerical value. While quality comprises multiple dimensions, a good estimator of an article's quality is user feedback. The latter can be explicit (rating over the quality of the articles (Lykourentzou, Papadaki, Vergados, Polemi, & Loumos, 2010; Weimer, Gurevych, & Muhlhauser, 2007), as performed in Wikipedia) or implicit (e.g. using quality indicators such as the time that the article remains unchanged (Blumenstock, 2008; Hu, Lim, Sun, Lauw, & Vuong, 2007)). In the present paper, we consider that the mechanism can estimate article quality as a single numerical metric. The way that this is measured (explicitly or implicitly) is out of the paper's scope and the interested reader is encouraged to refer to the above-provided references for details on the possible approaches. The mechanism compares each article's calculated quality to a pre-defined threshold. When the latter is not reached, the article needs to be enhanced. The selection of which user will be asked to contribute to which article is handled by the greedy allocation algorithm, presented in detail in the next section. The process of successive article contributions, quality evaluations and user selections continues until the article surpasses the quality threshold.

User selection process: A resource allocation problem

Resource allocation is a class of methods used to optimize a system's performance for one or multiple goals, under certain constraints, by assigning available resources to tasks (Patriksson, 2008). For example, suppose a factory that needs to manufacture a set of products and has a specific set of machines available for this purpose.

Each machine can complete the product manufacturing in a given amount of time. The allocation problem in this case is to identify which machine should process which product, so that the total production time is minimized.

For the case of the wiki system, machines can be mapped to wiki users and the products to the articles that are under the quality threshold. The resource allocation problem of this simple case would be: "Given a set of users and a set of articles that need enhancement, which user should be requested to contribute to which specific article, so that the average quality of articles inside the wiki is maximized?" (Figure 1)

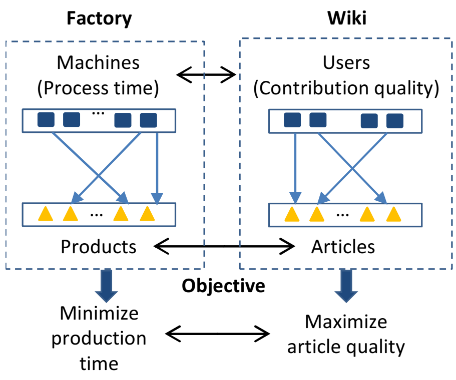

Fig.1. Mapping the resource allocation problem in the wiki system to product manufacturing.

In reality however, the problem of optimizing for quality at system-level inside a wiki has additional complexity compared to classical allocation problems:

1) First there is *uncertainty* regarding *resource availability*: users enter a wiki when they want, remain connected for as long as they like and they may or may not accept to make a contribution.
2) Secondly there is *uncertainty* regarding *resource capacity* (user contribution quality): the mechanism cannot know a priori if an expert user will enter the system, or whether the users that will enter are not knowledgeable enough to improve an article.
3) Third there is *uncertainty* regarding *task availability and state*: because users are free to accept or reject a system suggestion, as well to contribute to an article of their own choice, the quality level and the quality needs of the articles are not fixed but dynamically evolve with the system.
4) Fourth, a wiki article might require the contributions of multiple users before reaching the quality threshold, which brings upon the *need for sequenced, chain allocation*, with hands-off resource dependencies (the input of one user starts when another user has finished contributing to the same article).

The above constraints lead to a series of decisions. Because of the uncertainty in resource and task availability (points 1, 2, 3 above), we cannot assume a fixed pool of users or articles. Instead we need to suggest an article to a user at the moment he is available, according to the current needs of the system. Therefore, we propose using a heuristic greedy algorithm: once a user enters the wiki, the mechanism suggests him one article that the system estimates, at that moment, that he can provide quality enhancement. Finally, the sequenced, chain allocation nature of the problem (point 4) induces a high complexity. This indicates the need to work not in a long-term optimization horizon (e.g. calculating the optimal sequence of users to achieve high quality as fast as possible or save experts for future system needs), but rather with a one-step optimization goal (e.g. find the article that the specific user can improve the most, regardless of future system needs). As a result, the algorithm focuses on a one-step, localized optimization: it makes its decision based on the current state of the system (in terms of article quality needs) and does not optimize based on future need estimations.

The proposed algorithmic solution is designed especially for the constraints induced by the wiki quality optimization problem and does not necessarily seek optimality. However, our experiments have shown it could be sufficient to obtain significant quality enhancement of wiki articles (see Results section).

Resource allocation problem formulation

The overall article quality enhancement in wiki can be defined as a resource allocation problem formulated as follows. Given a set of:

- *Tasks*: the wiki articles, $A = \{i_1, i_2, \dots, i_{|A|}\}$, having quality below the threshold. Each article i is characterized by its quality q_i, which changes after a new user contribution, and a knowledge topic to which its subject is related, $D_i \in D = \{D_1, D_2, \dots, D_{|D|}\}$, where D is the set of knowledge topics of the wiki. Each article is considered to belong to exactly one knowledge topic for the sake of simplicity.
- *Resources:* the wiki users $U = \{u_1, u_2, \dots, u_{|U|}\}$. Each user u_j is characterized by the estimated quality improvement, $\widehat{\Delta q_k^{(J)}}$, he can bring to an article $k \in A$ in the wiki. As shown in (Lykourentzou, et al., 2010) $\widehat{\Delta q_k^{(J)}}$can be computed by the system as a function of: i) the topic D_i of the article, ii) the current article quality q_k and iii) the average quality improvements that user u_j has achieved in the past in the specific domain D_i: $\widehat{\Delta q_k^{(J)}} = f(D_i, q_k, \frac{\sum_{i=1}^{n} \Delta q_i^{(j)}}{n})$, where n is the total number of articles in topic D_i the user has contributed to. $\widetilde{\Delta q_k}$ is computed for each user, when he enters the system and for the whole set of articles that are below the quality threshold,

the problem is to find which article each user should be requested to contribute to, so that 1) the average quality of articles inside the wiki *C* is maximized:

$$C = \frac{\sum_{i=1}^{|A|} q_i}{|A|} \tag{1}$$

and 2) users are motivated to contribute. It is recognized that a prevalent factor for individual user participation in the case of wiki systems is self-fulfillment(Cho, Chen, & Chung, 2010; Nov, 2007). That is, the more positive effect a user's contribution has, the more satisfied the user is expected to be. Therefore, the problem also extends to maximizing the impact of the contribution of each user, i.e. maximizing:

$$\mathrm{I_j} = \frac{\sum_{\mathrm{i}=1}^{\mathrm{n}} \Delta \mathrm{q_i^{(j)}}}{\mathrm{n}}, \tag{2}$$

where Δq_i is the quality improvement that the contribution of user u_j brings to article i, and $1 < n < |A|$is the total number of contributions of the user.

Because of the need to satisfy: 1) a community-level objective (eq. 1) and 2) an individual level objective (eq. 2), this allocation problem is *multi-objective*.

The constraints considered in this problem formulation are:

- All articles need to surpass a given quality threshold T, $0 < T < 10$.

- A user can be recommended only one article at a time. The maximum number of articles that can be given to him is therefore a binary workload, $w = \{0,1\}$.
- Each user is recommended a specific article only once.
- Non-preemptive process. Once a user has been assigned to an article, he cannot be interrupted, to be recommended with another one.

Resource allocation algorithm – Rationale and development

The resource allocation algorithm needs to select the user whose estimated contribution can maximize the two objective functions defined in eq. (1) and (2). To find out which user should be selected we proceed as follows.

First we work on the community objective function. Suppose that user u_j is selected, accepts and edits article k. Then, the quality q_k changes, and subsequently the value of eq. (1) changes as follows:

$$\Delta C = \Delta\left(\frac{\sum_{i=1}^{n} q_i}{|A|}\right) \tag{3}$$

However, since the user action affects only article k, the above equation becomes:

$$\Delta C = \frac{\Delta q_k}{|A|} \tag{4}$$

Secondly, we work on the individual objective function. Suppose that user u_j edits article k and has already edited n articles. Then, according to eq. (2) the individual objective function, changes as follows:

$$\Delta I_j = \Delta\left(\frac{\sum_{i=1}^{n+1} \Delta q_i^{(j)}}{n+1}\right) = \left(\frac{\sum_{i=1}^{n} \Delta q_i^{(j)}}{n} + \Delta q_k^{(j)}\right) - \frac{\sum_{i=1}^{n} \Delta q_i^{(j)}}{n} = \Delta q_k^{(j)} \tag{5}$$

From eq. (4) and eq. (5) we can observe that both the community and individual objective functions are maximized by the user contribution that maximizes $\Delta q_k^{(j)}$. The above analysis shows that for the specific wiki optimization problem formulation the two objective functions (1) and (2) can be collapsed to one.

Finally, for a given user j, the article having the more chances to be enhanced while conserving user satisfaction is defined as ${k_j}^*, \widehat{\Delta q_{k^*}^{(j)}} = \max_{i=1..|A|} \widehat{\Delta q_i^{(j)}}\}$:

, where $|A|$ is the total number of articles below quality threshold in the wiki. As a result, the algorithm ranks articles for suggestion starting with the one that the user is estimated to improve the most. Because low-article quality articles leave more room for improvement, the algorithm gives natural priority to these articles, therefore providing a more balanced increase of quality of the wiki articles. Finally, the algorithm satisfies the constraints set above for: i) quality threshold ($q_i < T$), ii) user workload ($w = 0$) and iii) non-preemptiveness (user.busy = false). The final algorithm, in the form of pseudo-code, is presented in Table 1.

Table 1. The greedy wiki scheduling algorithm used

For every user that enters the wiki AND $w = 0$ AND user.busy=false
{
1. Find all articles with $q_i < T$, put them into list A
2. Sort A in descending $\widehat{\Delta q_i^{(j)}}$ order
3. Recommend to user the first article from A, remove it from list
4. If user accepts → give article to user, $w = 1$, user.busy=true → end ;
5. If user declines →
6.   If A non-empty → move to step 3
7.   Else → end;
}
* Users finishing an article are marked as non-busy (user.busy=false)

EVALUATION

The evaluation of our approach has been performed using a simulation model that has been parameterized from validated statistical analyses of the English Wikipedia (Ortega, 2009; Wu, Harrigan, & Cunningham, 2011). The model has been implemented into two simulation systems: i) one *benchmark* system reflecting a typical wiki, and a ii) a *smart* system extending the first one with the proposed coordination algorithm. The benchmark has been assessed against real-world behavior, on specific performance elements taken from large-scale analyses of Wikipedia found in the literature (Ortega, 2009; Urdaneta, Pierre, & Steen, 2009; West, Weber, & Castillo, 2012; Wu, et al., 2011), different from those used for the parameterization. Finally, we compared the performance of the benchmark system to that of the smart system, to examine whether the latter manages to provide better results.

Benchmark system

First we model the benchmark system, which represents the way that the wiki functions without the coordination mechanism. Users and articles are modelled coherently with the problem description (Methodology section):

User: The users are the set of wiki participants. They can view, compose and improve articles through edits. Each modelled user u_j is characterized by a:

- Domain expertise $\boldsymbol{e_{jd}}$, $j \in U, d \in D$. It represents the actual knowledge of the user u_j of the domain d, if this could be given by an objective oracle. It will be used to model the quality of the user's contribution on a given article. Every element in the expertise vector receives a value within the [0, 1] range, with 0 representing no knowledge, and 1 perfect knowledge. Each value of the expertise vector is initialized independently for each user through a random variable over an expertise distribution function *f*. The expertise of a user in a domain is considered to remain unchanged.

Article: Articles are characterized by two elements:

- The quality q, representing the actual quality of the article, is modeled as a single numerical value within the [0, 10] range. The initial quality of an article is zero (stub articles). The revision of an article changes its quality, according to eq. (8).

- Knowledge domain D_i. Each article in its initialization is randomly assigned to one article domain, according to a uniform distribution.

Functionality: Users arrive to the system following a Poisson process with arrival rate λ and they view an article. Taking into account that in typical wiki systems, the highly ranked articles tend to attract significantly more attention than lower-ranked ones (Kittur & Kraut, 2008; Zhang, Sun, Datta, Chang, & Lim, 2010) (an effect also referred to as preferential attachment pattern, observed in Wikipedia), the probability of each user viewing a specific article is modeled proportional to the exponent of the article's quality:

$$\mathrm{P}\{\mathrm{u_i \ views \ a_j}\} = \beta^{\,\mathrm{q_i}}, \tag{6}$$

where β is a view calibration factor.

After viewing an article, the user may choose to revise it and therefore change its quality, or leave without contributing. Since users are more likely to contribute to a subject that they know about, the probability of a user contributing to an article is modeled to be proportional to the user's expertise in the domain of the specific article:

$$\mathrm{P}\{\mathrm{u_i \ edits \ a_j} | \mathrm{u_i \ views \ a_j}\} = \kappa \cdot \mathrm{e_{id_j}}, \tag{7}$$

where e_{id_j} is the corresponding expertise of user i in the domain d_j of article a_j and κ is referred to as the contribution factor. The duration of each edit is a non-zero amount of time, randomly generated over a uniform distribution, during which no other interaction is allowed between the specific user and the system (maximum workload and non-preemptive process constraints). After the edit has been completed, the quality of the article changes, as follows:

$$\mathrm{q_{new}} = 10 \cdot \mathrm{e_{id_j}} + \rho \cdot \mathrm{q_{old}} - \rho \cdot \mathrm{e_{id_j}} \cdot \mathrm{q_{old}} \tag{8}$$

The rationale behind this equation is that users with higher expertise improve articles more than non-expert users, therefore quality is modeled as an increasing function of expertise. In parallel, the articles with low quality can be improved faster than high-quality ones, therefore the improvement of an article is modeled as a decreasing function of the old article quality:

$$\mathrm{q_{new}} - \mathrm{q_{old}} = 10 \cdot \mathrm{e_i^{d_j}} + \mathrm{q_{old}} \cdot \left[\rho \cdot \left(1 - \mathrm{e_i^{d_j}}\right) - 1\right], \tag{9}$$

where the first factor is fixed and the second part of the second factor is always negative. Finally, eq.(8) includes the possibility that a user contribution might worsen an article, according to a reduction factor $\rho \in [0,1]$. After a successful edit, the above process is repeated.

Smart **system featuring the coordination mechanism**

The smart system extends the benchmark with the coordination mechanism that suggests articles to users. Each user u_j is modeled to have a likelihood of acceptance a_j over the article propositions made by the coordination mechanism. Likelihood of acceptance receives a value in the [0,1] interval, with 0 and 1 representing respectively full rejection or acceptance. Furthermore, while the benchmark system simulates the actual behavior of users, the smart system relies on an *approximation* of user expertise. That is, to produce realistic results, before feeding the user expertise to the smart system we add random white noise. The span of the noise added is equal to the approximation error produced when neural networks are used to estimate wiki user expertise, i.e. ±0.3 in a scale of [0,10], as shown in (Lykourentzou, et al., 2010).

Simulation parameterization

The parameterization of the two simulators according to a realistic wiki functioning is realised using a generated sample of 1900 users and 20 knowledge domains, corresponding to a representative part of the English Wikipedia population as described by West et. al (West, et al., 2012). Based on the same study, the simulation time is set to 6000 simulation units, i.e. 300 days (20 units simulate 1 day), during which users produce 5000 articles.

The user arrival rate λ is determined by referring to the study of Urdaneta et.al (Urdaneta, et al., 2009), which, analyzing the English Wikipedia traffic, identifies a rate of approximately 800 page views/second, for a population of 1.8M users (West, et al., 2012). In our case, after normalization to the simulated day-to-time unit correspondence and population size, we obtain a λ parameter equals to $3.65 \cdot 10^{3}$, which corresponds to a user inter-arrival time of $2.741 \cdot 10^{-4}$ units.

For the contribution factor κ we refer to the study of Ortega (Ortega, 2009) who, performing one of the most extensive quantitative data analyses of the ten most prevalent Wikipedias (covering the years 2001-2007), finds that the number of different article contributions per author follows a Pareto distribution. Using the parameters provided by the aforementioned study for the case of the English Wikipedia, we have:

$$\kappa = \begin{cases} \alpha \cdot {\mathrm{x}_\mathrm{m}^\mathrm{a}}/{\mathrm{x}^{\mathrm{a}+1}}, & x > 1 \\ 0.0003, & 0 < x \leq 1 \end{cases}, \tag{10}$$

where x is the total number of articles the user has contributed to, $\alpha = 0.5836$ and $x_m = 1$.

The quality reduction factor ρ is set to 0.95. This assumes that users are not expected to deteriorate drastically the quality of an article, i.e. a community with low vandalism probability. This selection is confirmed by relevant studies, which point out that vandalism incidents in Wikipedia happen rarely (in approximately 5% of the total number of article revisions (Priedhorsky, et al., 2007)). Finally, the acceptance factor α is set to 0.03. This value is confirmed by the study of Coesleyet. al (Cosley, Frankowski, Terveen, & Riedl, 2007) who find that article recommendations in Wikipedia are accepted on average by 2.5-3.0% of their recipients, without significant differences among three different recommendation strategies examined. Finally, the expertise function f is modelled as a uniform product distribution of 3 independent random variables, uniformly distributed in the [0, 1] range. The specific distribution was chosen since it generates a population with a long expertise tail, i.e., many users with low to medium expertise and few with high expertise per domain, and this type of expertise distribution is very frequent for large-scale, general knowledge web applications such as Wikipedia.

Simulation Validation

Using the parameterisation detailed above, we obtain a benchmark system that presents similar behaviour to that of the English Wikipedia community. Figure 2 compares the output of the benchmark system with that of the real English Wikipedia community over two indicative factors: workload and quality. More specifically, Figure 2a illustrates the number of authors vs. their workload, i.e. the percentage of the total edits that they have undertaken, in normal and logarithmic view. We may observe that in both the benchmark system and in the data-driven reality plot most wiki authors undertake very few article edits and only a small percentage of users are likely to contribute to multiple articles. This "inequality of contributions" phenomenon is confirmed by many other studies in the field (e.g. (Lam & Riedl, 2009b)). Figure 2b illustrates the distribution of quality of the English Wikipedia in six quality categories, ranging from "starting" to "featured" articles, as reported in the study of Wu et. al (Wu, et al., 2011). In this case also we may observe that the benchmark simulator performs similarly to reality. We may also observe that whereas certain articles of very high quality exist (featured articles), most of the articles are of low quality, a characteristic often witnessed in the Wikipedia community.

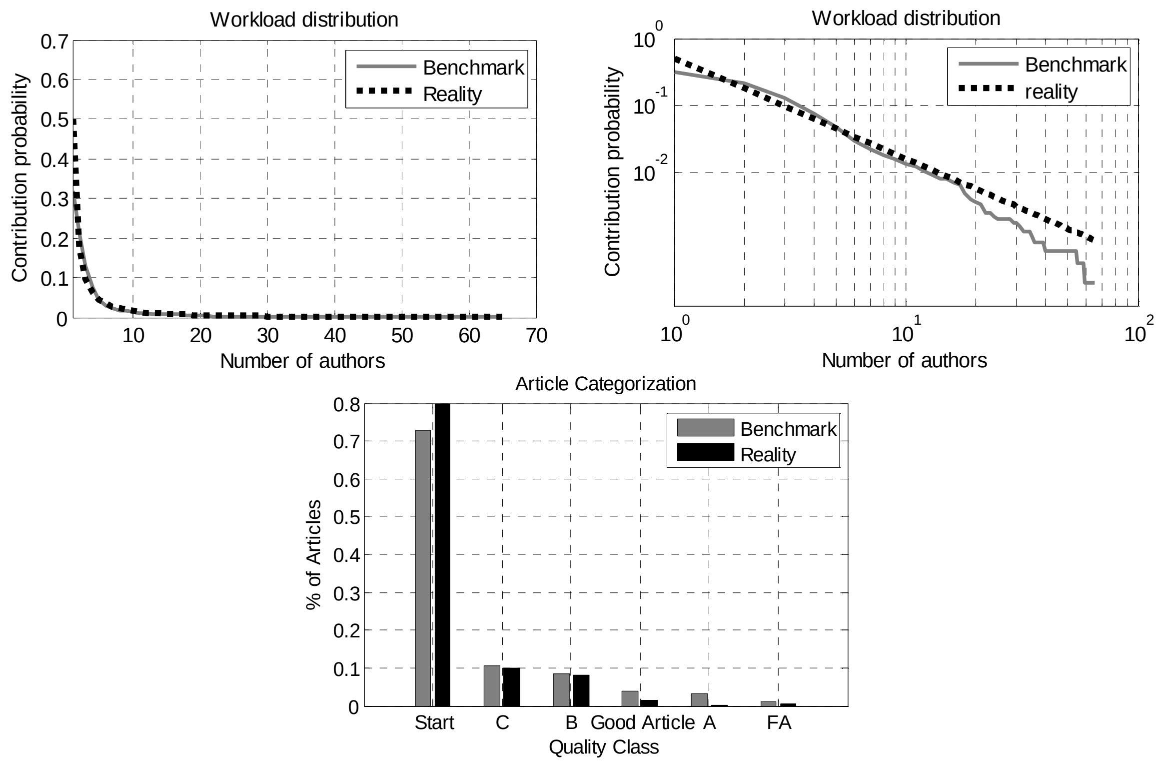

Fig.2. Model validation. The simulated benchmark system shows similar performance to the English Wikipedia data-driven reality, in terms of: a) workload per author in normal and logarithmic view and b) article categorization into quality classes.

RESULTS

First, we examine the level that the smart system meets the community objective defined in eq. (1), compared to the respective performance of the benchmark system (Figure 3). As one may observe, the average quality achieved, during the same time span, by the smart system is higher than the respective quality achieved by the benchmark system, indicating that the smart solution can better meet the objective set by the community. In addition, one may also observe that the smart approach meets the community objective in a timelier manner than this is met by the benchmark system.

These findings are important, because wiki articles often reach adequate quality levels in a slow manner and therefore an approach that could speed up this process would be particularly helpful. It should also be noted that the low average quality of the benchmark system, observed in Figure 3, can also be attributed to the large number of zero-quality articles (stub articles) that the system hosts, in parallel to the articles that do get edited. In contrast, the smart system having less zero-quality articles can achieve better average quality levels.

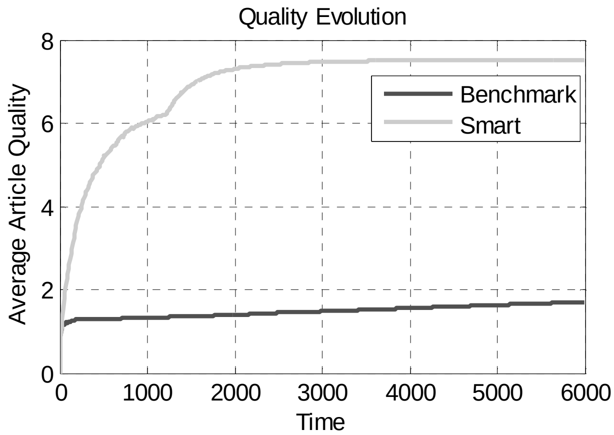

Fig.3. Evolution of the article quality achieved by the smart and the benchmark systems

Another interesting feature to examine is the distribution of the edits performed by participating users. As one may observe (Figure 4), the smart system results in slightly more edits per article, compared to the benchmark system, which is attributed to the fact that the system actively prompts users to contribute to articles.

This increase in user edits results however to a significant shift of the article quality distribution (Figure 5). In other words, the community manages to produce more qualitative articles through the use of the smart system, compared to the respective result achieved through the use of the benchmark system.

The above results indicate that the smart-enabled version of the wiki system can help increase the produced article quality, better allocate user skills and reduce the time needed for the articles to reach satisfactory levels of quality.

After examining the performance of the system for the basic scenario, in the following we elaborate on its output under different behavioral and environmental patterns of the involved user community. In this context, we examine system performance on two factors: user acceptance ratio and expertise availability. For both scenario variations, we will use the above basic scenario as our basis and change one behavioral factor at a time.

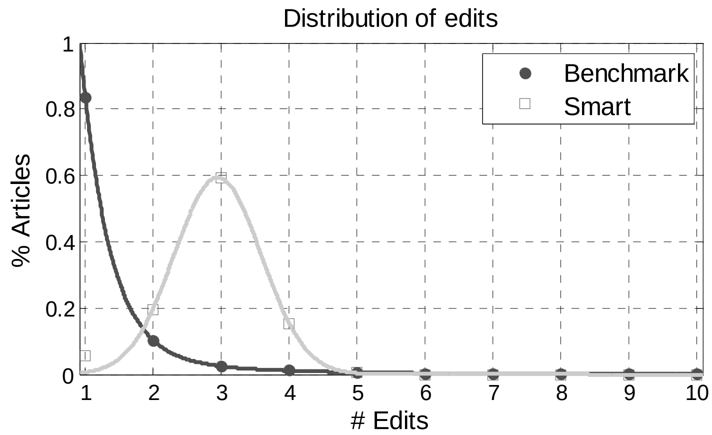

Fig.4. Distribution of the edits performed through the use of the benchmark and the smart systems

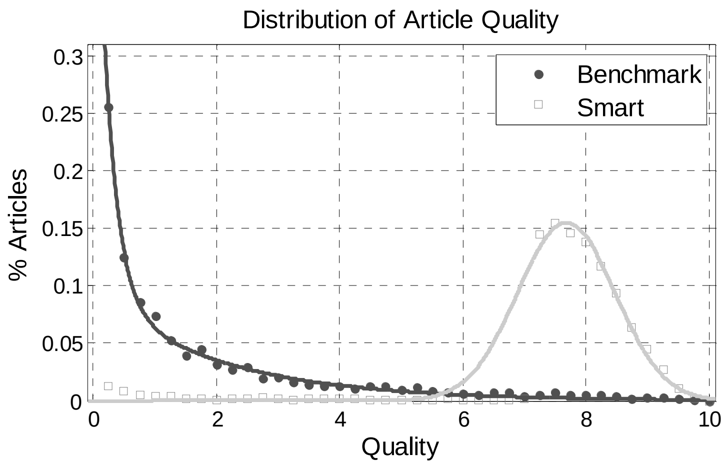

Fig.5. Comparison of the quality of the articles produced by the smart and the benchmark model

The effect of user acceptance ratio

User acceptance, to the suggestions of the system, is an user behavior element that may affect the performance of the coordination mechanism. User acceptance probability is linked to user interest over the subject of the suggested article, as well as on the overall motivation and knowledge sharing culture of the wiki community.

Given that the proposed mechanism makes its suggestions on the basis of user expertise, therefore implicitly on the basis on user interest on the domain of the article, the present scenario focuses more on investigating the effect that the different community motivation levels have over the capabilities of the algorithm. Figure 6 illustrates the performance of the smart system for different values of the α factor (likelihood of acceptance to smart system's suggestions), in the range of 0.1 – 10%. The performance of the benchmark system is also illustrated for comparison purposes, although the variation of factor α does not have an impact on the benchmark system's performance (users are not recommended to contribute to any articles in the benchmark system). As it can be expected, the smart system performance deteriorates as the probability of users accepting system requests falls, and it has performance similar to that of the benchmark system (dark grey line) for acceptance levels close to 0.1%.This result is important because it provides quantitative insight as to the quality gain that may be reached, or the loss that may be suffered, from higher or lower motivational levels of the participating community. Especially in cases where the knowledge sharing culture is not granted, e.g. corporate environments, the results illustrated in Figure 6 can be used to calculate the motivational level, and consequently the tangible or intangible rewards, which should be invested to enable the community reach the desired quality levels.

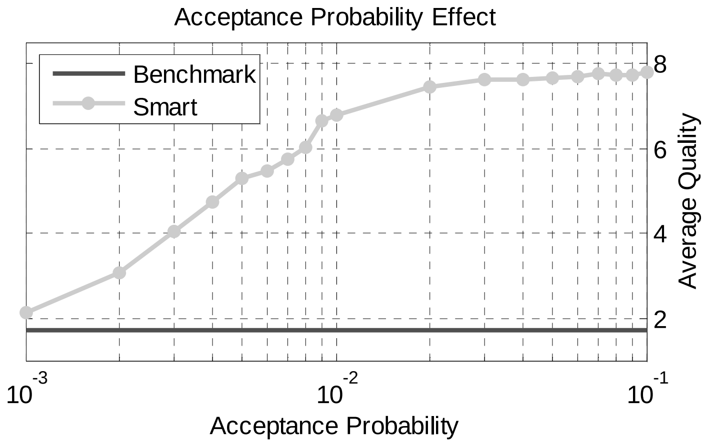

Fig.6. The effect of the population's acceptance probability over the quality performance of the smart system

The effect of expertise availability

The availability of experts is a second factor that plays a significant role over the knowledge production performance of the community. In this scenario variation, we investigate the impact of its effect, for different mean expertise levels, obtained by increasing the number of univariate factors of the expertise distribution function. Figure 7 presents the average quality and the average number of revisions achieved by the benchmark and the smart systems. We may observe that, as it can be expected, quality increases with the availability of expert resources for both systems. However the smart system manages at all times to achieve significantly higher quality levels compared to the benchmark. We may also view that even in very low mean expertise availability levels, the average article quality achieved by the smart system is still satisfactory (higher than 6 in a scale of 10). Regarding the average number of revisions we may observe that, for the benchmark system, it remains relatively steady and it increases slightly only for high expertise levels, as a result of the effect that expertise has on edit probability. The behavior of the smart system is somehow different. For low expertise availability levels, the number of revisions is significantly higher, because of the active effort that the system makes to ensure high quality levels. As mean expertise increases, this is no longer the case – the smart system can more easily identify experts and recommend them the articles that need enhancement. This tradeoff between quality and revisions ceases for expertise levels above 0.4 (in the scale [0 1]). For these levels we may see that the smart system actually achieves higher quality and lower number of required revisions at the same time, compared to the benchmark system. Therefore for a wiki community the members of which have medium to high expertise, in one or more domains, the smart system can simultaneously improve both the collective quality objective and the effort required to reach it. Finally, the output examined in Figure 7, provides designers with an expectancy of the capabilities and limitations of both systems, as well as with an estimation of the number of additional expert resources needed in order to reach desired quality or revision levels.

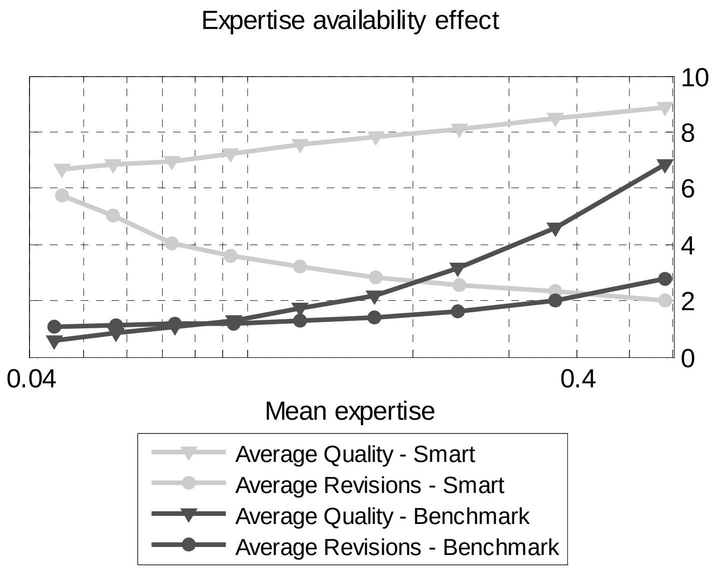

Fig.7. The effect of the population's mean expertise levels over the performance of the benchmark and the smart systems

DISCUSSION AND PERSPECTIVES

The results reported in this article show the usefulness of crowd coordination in wikis in globally increasing the quality of produced articles. The model used is designed from field studies on the English Wikipedia and manages to successfully reproduce its behavior, therefore constituting a sufficient platform for the needs of this work. Nevertheless, additional issues can be considered, which will deepen our understanding on the use of coordination mechanisms for wikis and provide perspectives for further research. These issues are discussed in the following.

Article prioritization

The coordination mechanism used in this work targets for a balanced quality improvement across the wiki, and therefore assumes that all articles have the same priority. Nevertheless it may be the case that not all articles value the same for the involved wiki community. Knowledge value quantification means (e.g. (King, Barlatier, Naudet, Vidou, & Watrinet, 2009)) or using interest indicators, such as the length of discussion pages or the number of users talking about an article, could in this sense be helpful in asserting this. Our mechanism can also be used to provide article interest indicators, on the basis that uninteresting articles will be refused by the majority of requested users. Even when article value or usefulness is estimated, prioritizing articles still remains a decision to be taken depending on the wiki operational objectives. This problem is related to diversity handling: how to consider articles in the long tail (Anderson, 2006) distribution of valuable articles. Since the self-fulfillment of users is essential to ensure participation in the wiki, articles of the long tail should be proposed with explanations of a clear objective.

Semantic domain similarity

The model used in this paper assumes that articles belong to exactly one knowledge domain, and therefore that expertise on a specific domain can only be estimated for users that have made at least one contribution to that domain. Domains however may have a semantic similarity. This fact could, under conditions, also imply that expertise in similar domains may be deducted for users that have contributed to semantically close domains. Therefore, an extension of the coordination mechanism's design to this direction could significantly increase the availability of experts that the mechanism considers, and be particularly helpful for low-population wikis or domains with few contributions.

Impact of recommendations in wikis

Wikis are systems that depend on user participation. As such, they use an open collaboration model where users spontaneously review and contribute to articles. Therefore, the impact of recommendations on the behavior of users participating in a wiki, and in other social mass participation systems in general, is an open question. Since this bias has yet to be determined, allowing for a mixed model of article suggestions and ad-hoc contributions, like the one employed by the coordination mechanism of this paper (where users are given recommendations, are free to accept or reject them and can contribute to any article they like) is advisable. Furthermore, accompanying each article recommendation with an explanation as to the system-level impact that the specific user's contribution can have, may also improve the understandability of the recomendations and minimize the risk of negatively affecting user participation.

Article quality evaluation bias

The quality evaluation of an article may be biased, especially when explicit feedback (user ratings) are used or the article is controversial. That is, a small group of users could well "join forces" to artificially lower or improve the estimated quality of an article, thus accelerating or stopping the active efforts of the algorithm for more contributions. This is for example a well-known case in other social-involving domains (such as Youtube regarding the visibility of posted videos). A real-world implementation of the mechanism would therefore reveal the extent to which such trust handling issues happen and what preventive measures, tailored to the specific community, can be taken.

CONCLUSIONS AND FUTURE WORK

In view of the importance that content quality has for mass-participation, collaborative knowledge creation efforts on the Web, in this paper we present an algorithm-based crowd coordination mechanism, which aims at guaranteeing a global article quality inside a wiki. The proposed mechanism uses resource allocation to match users with wiki articles, in such a way as to satisfy the demand for maximized average article quality and improved user contribution impact. Experimental results, obtained through simulation modeling, parameterized and validated through Wikipedia analyses, indicate that the proposed coordination mechanism can improve overall quality, and be resilient in this result for differentiated values of two important environmental parameters: user acceptance and expert availability. Additional open issues (article diversity handling, impact of article suggestions in wikis, etc.) and their implications for further research are also discussed.

Future work includes extending the proposed coordination mechanism to additional communities, such as corporate wikis, where the knowledge sharing culture and rewards are expected to play a significant role in the algorithm design. Additional constraints, such as article deadline could be added, necessitating modifications of the problem formulation presented in this paper and of the resource allocation strategy. Finally, it would pose significant research interest to extend the wiki problem, tackled in this paper, as a generic people-to-task allocation problem (including systems such as Q&A sites, crowdsourcing and crowdfunding), and to examine whether analytical solutions for this could be found.

REFERENCES

1. Anderson, C. (2006). *The Long Tail: Why the Future of Business Is Selling Less of More*: Hyperion.
2. Barberio, V., & Lomi, A. (2009, July 19-24). *Weighting Structures: Evolutionary Dynamics of Innovation Networks in Virtual Communities.* Paper presented at the Proceedings of the 3d International Conference on Online Communities and Social Computing: Held as Part of HCI International 2009, San Diego, CA.
3. Bernstein, M. S., Karger, D. R., Miller, R. C., & Brandt., J. (2012). Analytic Methods for Optimizing Realtime Crowdsourcing. . *CoRR, abs/1204.2995*.
4. Blumenstock, J. (2008). *Size matters: word count as a measure of quality on wikipedia*. Paper presented at the WWW '08: Proceeding of the 17th international conference on World Wide Web, Beijing, China.
5. Butler, B., Joyce, E., & Pike, J. (2008). *Don't look now, but we've created a bureaucracy: the nature and roles of policies and rules in wikipedia*. Paper presented at the Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems.
6. Cho, H., Chen, M., & Chung, S. (2010). Testing an integrative theoretical model of knowledge-sharing behavior in the context of Wikipedia. *Journal of the American Society for Information Science and Technology, 61*(6), 1198-1212.

7. Cosley, D., Frankowski, D., Terveen, L., & Riedl, J. (2007, January 28 - 31). *SuggestBot: using intelligent task routing to help people find work in wikipedia.* Paper presented at the Proceedings of the 12th international conference on Intelligent user interfaces, Honolulu, Hawaii, USA.
8. Denning, P., Horning, J., Parnas, D., & Weinstein, L. (2005). Wikipedia risks. *Commun. ACM, 48*(12), 152-152.
9. Ghosh, A., & Hummel, P. (2012). *Implementing optimal outcomes in social computing: a game-theoretic approach*. Paper presented at the Proceedings of the 21st international conference on World Wide Web.
10. Hoffmann, R., Amershi, S., Patel, K., Wu, F., Fogarty, J., & Weld, D. S. (2009, April 4-9). *Amplifying community content creation with mixed initiative information extraction.* Paper presented at the Proceedings of the 27th international conference on Human factors in computing systems, Boston, MA, USA.
11. Hu, M., Lim, E.-P., Sun, A., Lauw, H. W., & Vuong, B.-Q. (2007, November 06 - 10). *Measuring article quality in wikipedia: models and evaluation.* Paper presented at the Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, Lisbon, Portugal.
12. King, W. R., Barlatier, P.-J., Naudet, Y., Vidou, G. r., & Watrinet, M.-L. (2009). Valuing Knowledge Within Virtual CoPs: The Quest for Meaningful Indicators *Knowledge Management and Organizational Learning* (Vol. 4, pp. 253-277): Springer US.
13. Kittur, A., & Kraut, R. E. (2008, November 8 -12). *Harnessing the wisdom of crowds in wikipedia: quality through coordination.* Paper presented at the Proceedings of the 2008 ACM conference on Computer supported cooperative work, San Diego, CA, USA.
14. Kittur, A., Lee, B., & Kraut, R. E. (2009, April 4 - 9). *Coordination in collective intelligence: the role of team structure and task interdependence.* Paper presented at the Proceedings of the 27th international conference on Human factors in computing systems, Boston, MA, USA.
15. Kong, N., Hanrahan, B., Weksteen, T., Convertino, G., & Chi, E. H. (2011, February 13 - 16). *VisualWikiCurator: human and machine intelligencefor organizing wiki content.* Paper presented at the Proceedings of the 16th international conference on Intelligent user interfaces, Palo Alto, CA, USA.
16. Lam, S. K., & Riedl, J. (2009a). *Is Wikipedia growing a longer tail?* Paper presented at the Proceedings of the ACM 2009 international conference on Supporting group work.
17. Lam, S. K., & Riedl, J. (2009b, May 10 - 13). *Is Wikipedia growing a longer tail?* Paper presented at the Proceedings of the ACM 2009 international conference on Supporting group work, Sanibel Island, Florida, USA.
18. Liu, J., & Ram, S. (2011). Who does what: Collaboration patterns in the wikipedia and their impact on article quality. *ACM Trans. Manage. Inf. Syst., 2*(2), 1-23.
19. Louridas, P. (2006). Using Wikis in Software Development. *IEEE Software, 23*(2), 88-91.
20. Lykourentzou, I., Papadaki, K., Vergados, D. J., Polemi, D., & Loumos, V. (2010). CorpWiki: A self-regulating wiki to promote corporate collective intelligence through expert peer matching. *Inf. Sci., 180*(1), 18-38.
21. Nath, S., Zoeter, O., Narahari, Y., & Dance, C. (2011). *Dynamic mechanism design for markets with strategic resources*. Paper presented at the The Twenty-Seventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-11), Corvallis, Oregon.
22. Nov, O. (2007). What motivates Wikipedians? *Commun. ACM, 50*(11), 60-64.
23. O'Mahony, S., & Ferraro, F. (2007). The emergence of governance in an open source community. *Academy of Management Journal, 50*(5), 1079–1106.
24. Ortega, F. (2009). *Wikipedia: A quantitative analysis.* Unpublished Doctoral Thesis, Universidad Rey Juan Carlos, Móstoles, Spain.
25. Patriksson, M. (2008). A survey on the continuous nonlinear resource allocation problem. *European Journal of Operational Research, 185*(1), 1-46.
26. Priedhorsky, R., Chen, J., Lam, S. K., Panciera, K., Terveen, L., & Riedl, J. (2007). *Creating, destroying, and restoring value in wikipedia*. Paper presented at the Proceedings of the 2007 international ACM conference on Supporting group work.
27. Psaier, H., Skopik, F., Schall, D., & Dustdar, S. (2011, Aug. 29 2011-Sept. 2 2011). *Resource and Agreement Management in Dynamic Crowdcomputing Environments.* Paper presented at the Enterprise Distributed Object Computing Conference (EDOC), 2011 15th IEEE International.
28. Urdaneta, G., Pierre, G., & Steen, M. v. (2009). Wikipedia workload analysis for decentralized hosting. *Comput. Netw., 53*(11), 1830-1845.

29. Weimer, M., Gurevych, I., & Muhlhauser, M. (2007, June 25 - 27). *Automatically assessing the post quality in online discussions on software.* Paper presented at the Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions (ACL '07), Prague, Czech Republic.
30. Welser, H. T., Cosley, D., Kossinets, G., Lin, A., Dokshin, F., Gay, G., et al. *Finding social roles in Wikipedia*. Paper presented at the Proceedings of the 2011 iConference.
31. West, R., Weber, I., & Castillo, C. (2012). *Drawing a Data-Driven Portrait of Wikipedia Editors*. Paper presented at the WikiSym'12.
32. Wu, G., Harrigan, M., & Cunningham, P. (2011). *Characterizing Wikipedia pages using edit network motif profiles*. Paper presented at the Proceedings of the 3rd international workshop on Search and mining user-generated contents.
33. Yu, C., Lakshmanan, L., & Amer-Yahia, S. (2009). *It takes variety to make a world: diversification in recommender systems*. Paper presented at the Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology.
34. Zhang, Y., Sun, A., Datta, A., Chang, K., & Lim, E.-P. (2010). *Do Wikipedians follow domain experts?: a domain-specific study on Wikipedia knowledge building.* Paper presented at the Proceedings of the 10th annual joint conference on Digital libraries, Gold Coast, Queensland, Australia.