International Journal of Computer and Information Technology (ISSN: 2279 – 0764)
Volume 03 – Issue 06, November 2014

Incentives for Online Communities

David DeAngelis
The Center for Identity
The University of Texas at Austin
Austin, TX, USA

K. Suzanne Barber
The Center for Identity
The University of Texas at Austin
Austin, TX, USA
Email: sbarber {at} identity.utexas.edu

Abstract—**Online communities promote wide access to a vast range of skills and knowledge from a heterogeneous group of users. Yet implementations of various online communities lack consistent participation by the most qualified users. Encouraging such expert participation is crucial to the social welfare and widespread adoption of online community systems. Thus, this research proposes techniques for rewarding the most valuable contributors to several classes of online communities, including question and answer (QA) forums and other content-oriented social networks. Overall, novel quantitative incentives can be used to encourage their participation. Using a game theory approach, this research designs and tests an incentive mechanism for QA systems. Based on survey data gathered from online community users, the proposed mechanism relies on *systemic rewards*, or rewards that have tangible value within the framework of the online community. This research shows that human users have a strong preference for reciprocal systemic rewards over traditional rewards. Furthermore, this research shows that it is possible to motivate participation from the most valuable contributors to an online community.**

Keywords-incentives, online communities, expert participation, game theory

I. Introduction

Untapped capabilities permeate large-scale networks. Search engines specialize in identifying existing static documents on a network that are appropriate for a given query. Online communities such as QA forums, discussion forums, social networks, and news aggregators provide a method of connecting users and resources that can leverage both the static and dynamic (i.e., live) capabilities of a network of human users. Online communities promote wide access to a vast range of skills and knowledge from a heterogeneous group of users. Yet current implementations of various online communities lack consistent participation by the most qualified users. [1]

Online communities are enabled by the prevalence of popular websites built upon social technology. Websites built on user-generated content (UGC) are prevalent, and the perceived value of this content is growing rapidly [2]. Sites like Twitter, Yelp, Digg, Reddit, eBay, Yahoo! Answers, Amazon, and many others rely on content created by their users, whether product reviews and descriptions, restaurant suggestions, movie recommendations, or any other kind of information. Often, these websites allow each user to create an online identity. Through contributions to the site, users build a reputation through the collective whole of other users. This reputation and its associated measure of trust form the essence of an online community.

For these communities to work successfully their reward systems must identify and access experts in any given area, connect the responder(s) to the original questioner, and motivate potential responders to participate. [1, 3, 4]. Thus, we focus on identifying *content creators*, or experts with desired knowledge, and motivating them to contribute to the social benefit of the online community. No single user has complete knowledge across many different domains. On a large enough network, however, it is likely that somebody has expertise in nearly every domain.

Many different online communities, including QA systems, are currently available, but they all suffer from a similar set of problems. First is a lack of participation. It is beneficial to encourage expert participation from users in order to reliably secure valuable content, thus directly adding value to the community [1, 3, 4]. A major barrier toward participation is the time investment needed by a content creator to find an appropriate piece of content to create or an appropriate question to answer. A second problem with current online communities is a lack of confidence in the expertise and trustworthiness of the content creators. A third problem is that online communities suffer from various social phenomenon such as nepotism, reciprocity, and bandwagon effects.

Overall, encouraging expert participation is crucial to the social welfare and widespread adoption of online community systems. Thus, we propose techniques for developing incentive mechanism to motivate the most valuable contributors to several classes of online communities, including question and answer (QA) forums. This research represents an investigation into QA systems, while the major findings are widely applicable to other content-oriented social networks. Specifically, this research makes the following contributions:

- We examine meaningful rewards for online community participation.
- We design and implement an influence-based incentive mechanism to encourage the strongest expert participation.

— We illustrate how novel quantitative incentives can be used to encourage expert participation in online communities.

II. Mechanism Design

Game theory is relevant to the aforementioned problems in QA systems because users do not always answer questions to the best of their ability, yet they are typically rational actors in competition for limited resources. Mechanism design is a particular field of game theory, originally developed for applications to economics. These applications include auctions, markets, pricing strategies, and many others.

Game theory traditionally focuses on careful observation and strategy surrounding the behavior of rational agents in a given competitive game. Mechanism design can be considered a form of reverse game theory; the game is designed to induce certain strategies and therefore certain outcomes. Like Parkes [5], Tuomas Sandholm defines mechanism design as: "Mechanism design is the art of designing the rules of the game so that a desirable outcome is reached despite the fact that each agent acts in his own self-interest" [6].

Each user, or rational agent, has a *type* (θ), which represents their full capabilities and expertise. From their contributions to the QA system, we can see their reported type, or $\hat{\theta}$. A *social choice function* maps the true type θ to an outcome, whereas a mechanism maps the reported type $\hat{\theta}$ to an outcome. The goal of mechanism design is to design a game that has an equilibrium state that implements the social choice function.

Game theory is relevant to the aforementioned problems in QA systems because users do not always answer questions to the best of their ability, yet they are typically rational actors in competition for limited resources. Mechanism design is a particular field of game theory, originally developed for applications to economics. These applications include auctions, markets, pricing strategies, and many others.

Game theory traditionally focuses on careful observation and strategy surrounding the behavior of rational agents in a given competitive game. Mechanism design can be considered a form of reverse game theory; the game is designed to induce certain strategies and therefore certain outcomes. Like Parkes [5], Tuomas Sandholm defines mechanism design as: "Mechanism design is the art of designing the rules of the game so that a desirable outcome is reached despite the fact that each agent acts in his own self-interest" [6].

Each user, or rational agent, has a *type* (θ), which represents their full capabilities and expertise. From their contributions to the QA system, we can see their reported type, or $\hat{\theta}$. A *social choice function* maps the true type θ to an outcome, whereas a mechanism maps the reported type $\hat{\theta}$ to an outcome. The goal of mechanism design is to design a game that has an equilibrium state that implements the social choice function.

Essentially, the problem of designing an incentive mechanism for QA systems and other participatory online communities can be expressed in three steps:

(1) Identify a set of desired outcomes;

(2) Select a reward that is meaningful to the target audience;

(3) Distribute the reward in a fashion that maintains the desired outcomes.

A. *Identifying Desired Outcomes*

In this research, mechanism design is driven by the assumption that expert responders are highly desirable in a QA system. *Expertise* is defined as the ability of a user to answer a given question to the satisfaction of the questioner. Given a question, how can the pool of potential answerers be indexed and searched to predict who is capable of and willing to provide an answer. Estimating which user is most likely to give a satisfactory answer is a challenging problem and requires a complex model of human expertise along:

— *Expertise dimensions*: the various distinct areas of human knowledge and the expert's ability in each of these areas;

— *Compatibility*: the likelihood that the answerer's personality and approach to answering questions matches that of the questioner; and

— *Willingness*: the probability that the answerer will be willing to invest the time required to answer the question.

In QA systems, expertise can be identified through a *multi-dimensional topic-specific expertise model* [7, 8]. Essentially these techniques draw from two sources of information: content-based information and link-based information. Content-based expertise identification techniques involve analyzing the text content generated by users in an attempt to ascertain the skill level of the user. This has been done in the context of clustering documents into topics by [9]. Additionally, Zhang et al. [10] have developed the QuME algorithm, which identifies expertise based on matching keywords. *Link-based expertise identification techniques* use the underlying graph structure generated from the social interactions of the users within the community. Link information can be used to identify expertise nearly as well as human raters [11]. And by applying a variation of the HITS algorithm to the much broader Yahoo! Answers data set has been used to identifying authoritative users, or expert responders [12, 13, 14]

Before a mechanism can be created, it is imperative to fully understand the desired outcome of a game. In this research, the game involves expert participation in QA systems. Thus, a full understanding of desired outcomes is necessary to develop the rules that form this incentive mechanism. The following is a list of proposed desirable outcomes in a QA system:

— Users are not penalized for asking a question or giving a poor answer.

— Satisfactorily answering a question yields a greater reward than unsatisfactorily answering that question.

— Satisfactorily answering a question of higher value (asker has more influence) should yield higher rewards than answering a question of lower value.

— Users who answer very difficult questions should be rewarded for doing so.

— Similarly, an endorsement from a user with high influence should yield a higher reward than one from a user with lower influence.

— Recently added users should be able to earn a meaningful amount of influence in a reasonable time in order to compete with more established users.

— All users should be rewarded for other forms of participation such as endorsing or denouncing posted content, including questions and answers.

This list of outcomes captures the desired behavior of users interacting on the proposed QA system. Some of these desired outcomes may sound counterintuitive, particularly the first one. A user should not be penalized for asking a question because part of the value of a QA system is having a rich corpus of questions and answers readily accessible as a reference. Discouraging asking questions reduces value for the questioner who is seeking answers and also for users who would benefit from answering the question. Poor answers made in earnest should also not be discouraged. It is important to encourage participation, and the cost of ignoring or filtering poor answers in minimal.

B. Selecting Rewards

While expertise models are useful for recommending questions and inducing suggested behavior, the incentive mechanism is used for rewarding actual observed beneficial behavior. These two techniques function as a push and a pull serving as separate means toward the same end. An incentive mechanism is an algorithm that must encourage optimal system-wide behavior from self-interested agents.

Various incentive mechanisms have been used in online communities to encourage participation. Fundamentally, an incentive mechanism rewards a user who exhibits socially beneficial behavior by giving him/her something of value. It is increasingly common for online communities to use *achievement-based incentives* to motivate users to participate. Such incentives include leader-board standings, custom titles, trophies, and avatars. This is essentially giving virtual prizes for participating in the online community. Yahoo! Answers and many similar sites reward contributors with arbitrary *points* for positive contributions. Additionally, they may offer a leader board position, or virtual trophies and badges. Such incentives are very effective for a portion of the population. The leading users on Yahoo! Answers will often answer 80+ questions per day, every day, for as long as the website has been live [15]. These fanatical users are strongly motivated by the achievement-based incentives. Yet for many users of such systems, these rewards are meaningless.

A *reciprocal incentive* is fundamentally different. A reciprocal incentive rewards people who answer questions in a QA website by assigning them a score, which is then used to calculate the reward that another person will receive when answering the first person's questions. Therefore, people who answer the questions of others will be given priority when asking questions of their own. This is like gaining priority access to the knowledge of the entire community in exchange for providing answers. The key difference is that the rewards in a reciprocal incentive mechanism have systemic value. These rewards directly help the recipient accomplish something within the system. An achievement reward must provide its own value in isolation to the recipient.

This research attempts to identify a reward that is valuable to a larger portion of the users. Such meaningful rewards are called *systemic rewards* because they add value within the framework of the system. They are designed to give extra functionality, enjoyment, or ease of use to the awardee. For example, in a QA system a potential systemic reward is priority access to potential responders when asking a question. This reward is said to be *reciprocal* because the reward gained by a content creator is a function of the social influence of the user who requested the content. In the context of QA, a question asked by a user in high standing is “worth more” than a question asked by a user in lower standing.

Furthermore, effective systemic reciprocal rewards can be used for encouraging expert participation in an online community. This research posits that a reciprocal incentive will encourage greater participation, particularly from experts, than an achievement incentive in an online community such as a QA forum. The proposed QA system is designed around the assumption that people have varying levels and areas of expertise, but everyone has something valuable to contribute. Therefore reward is based on both demonstrated expertise and participation.

A required step in creating such a mechanism is to decide what reward to give beneficial users. In order to address this question of what type of reward to give and to gauge human interest in reciprocal mechanisms, a short web-based survey was administered to 380 anonymous volunteers, including mostly engineering graduate students. No personally identifiable information was collected, and the test subjects were not compensated for their participation in any way. The survey link was emailed through several distribution channels, and participants were encouraged to spread the link to others.

Survey results confirm that the majority of the survey respondents are familiar with QA websites and gain value from them by looking up existing question and answer pairs. It is suspected that a large portion of the reference usage was driven by search engine query results. Only roughly one quarter of the responders have used QA systems in an active sense; that is, they have asked or answered questions. It is likely that this is due to a lack of suitable rewards for participation.

In addition, 64% of survey responders prefer reciprocal incentives to achievement incentives. Differences between the two incentive types are statistically significant according a binomial test with $\alpha = 0.01$. This shows us that approximately 75% of the responders claim that an achievement incentive motivates them to participate only a little or none at all. In comparison this number for the reciprocal mechanism is only 55%. Furthermore, nearly twice as many people rated the reciprocal incentives as having some or a lot of effect as compared to the achievement incentive.

In practice, using a reciprocal incentive does not preclude using additional achievement-based incentives. Many successful online communities function very well with achievement incentives. This survey simply indicates that there is strong interest in creating something different. There is no reason that a community cannot be built using both reciprocal and achievement based incentives in order to appeal to the largest target population.

Based on these survey results, we have established that reciprocal incentives are preferred over achievement incentives. With this information we can construct an incentive mechanism based on reciprocal systemic rewards for encouraging expert participation in an online community.

C. Distributing the Reward

The difficulty with mechanism design is mapping desired outcomes to a set of rules for distributing rewards that enforces these outcomes with self-interested agents. Once a reward is chosen, it must be carefully distributed to the users in order to encourage positive behavior and discourage negative behavior. Such a mechanism is said to be *incentive compatible*. Some examples of beneficial responder behavior in QA include the creation of prompt, relevant, and correct (where applicable) responses. Examples of beneficial questioner behavior include asking rich questions, such as asking for advice pertaining to a detailed situation. One of the greatest strengths of QA is that humans are capable of answering more sophisticated questions than those most suited to an internet search query. The incentive mechanism must be designed to encourage such beneficial behaviors.

In addition to encouraging beneficial behavior, the incentive mechanism must discourage harmful behaviors. First, false or biased responses to questions can cause harm and should be minimized. Second, spam is a major concern for internet-based QA systems. While perhaps not as harmful as false information, unintelligible responses are of no use. This could be caused by something as straightforward as language issues, or unintelligibility can be symptomatic of something deeper, such as a large disparity of expertise between the questioner and the responder for the question topic. A third point of concern for many QA systems is each user's question to answer ratio. A forum full of questions with very few answers is of little use. Likewise, a QA system with too few questions is underutilizing the skills of its user base. This research does not propose discouraging questions; it proposes encouraging strong responses.

The analysis in this section assumes that the reciprocal systemic rewards are desired and that the utility of these rewards is linear with respect to reward quantity. Furthermore, this section introduces notation and terminology originally developed by Hurwicz and Reiter [16] for designing economic mechanisms and adapted here to QA systems. Question and answer systems can be considered privacy-preserving games of private information, or Bayesian games. Users, or agents, generally know their own expertise, but that is not necessarily public information; it is considered private. Users are not forced to answer questions or share their full knowledge, though they can choose to, hence the game is privacy-preserving. Each user, or agent, in a QA system is capable of answering questions honestly, promptly, and to the best of its ability. This optimal set of behaviors is called the agent's true *type* and is represented with the symbol θ. A user is said to *report* its type by expressing certain behaviors. The observed actions of agents are then called the reported type, represented as $\hat{\theta}$.

The set of all possible types, or behaviors, that a user can take, including asking questions, answering questions, evaluating content, and defecting from the system is called Θ. The mechanism y is a set of rules, or a function, that takes into account the game environment, g, and is executed on a reported type, $y(\hat{\theta})$. The result of this mechanism is an allocation of influence points, or a particular *outcome*, z in the set of all outcomes Y. Therefore:

$$y(\hat{\theta}) : \Theta \rightarrow Y$$

The goal of mechanism design is to design an allocation y based on reported type $\hat{\theta}$ that has an equilibrium state ξ that implements the social choice function $f(\theta)$. The social choice function can be considered a target benchmark for the mechanism. $f(\theta)$ maps the true type θ, not the reported type $\hat{\theta}$ which can include deception or fraud, of each agent to the set of desired outcomes, X.

$$f(\theta): \Theta \rightarrow Y$$

The revelation principle states: "For any Bayesian Nash equilibrium there corresponds a Bayesian game with the same equilibrium outcome but in which players truthfully report type" [17]. An incentive mechanism is designed to operate in a particular equilibrium state. At equilibrium, agents report their type as a function of their true type, $\hat{\theta}(\theta)$. Searching for equilibria in a Bayesian game is very difficult because the action space for each agent is large. It can choose to answer questions honestly, or it can lie, or it can refuse to respond at all. The revelation principle allows us to restrict our search to just those states where agents truthfully report their type, $\hat{\theta} = \theta$. In other words we must consider only states in which agents are honest, albeit selfish participants.

Consider the aforementioned set of desired outcomes and these rules:

(1) Users earn influence points when answering questions correctly.

(2) Answers deemed incorrect or spam do not receive a reward.

(3) The influence points earned by a user answering a question are dependent on the influence of the user who asked the question.

(4) An influence point bonus is awarded for authoring the best answer to a question, and this bonus is also dependent on the influence of the user who asked the question.

(5) Users have a nonzero influence point balance when entering the system.

(6) Influence points decay with time.

The first desired outcome is that users are not penalized for asking a question or giving a poor answer. There is no penalty for these behaviors specified in the rules. The only penalty is implicit. That is, users will waste their own time and effort by giving poor answers or asking worthless questions. Closely related to this is the second desired outcome that satisfactory answers are worth more than unsatisfactory answers. According to Rule 2, unsatisfactory answers are not rewarded.

The third desired outcome is that answering a question of higher value yields higher rewards. Rule 3 supports this outcome because users who answer higher value questions correctly receive greater rewards than those who answer lower value questions. These higher value questions are those that originate from highly influential users.

Rule 4 enforces the fourth desired outcome that users should be rewarded for answering very difficult questions in place of ordinary questions. Very difficult questions do not necessarily have a larger reward associated with them because rewards are based on questioner influence, and more influential users do not necessarily ask more difficult questions [15]. This concern is why a “best answer” bonus is built into this mechanism. A more difficult question is likely to draw fewer answers, increasing the chances of giving the designated best answer and earning the bonus reward.

Rules 5 and 6 support the fifth desired outcome that recently added users should be able to compete with more established users. Returning users may build up substantial influence through participation over time. A new user is likely to have very little accumulated influence even though he may have significant expertise. This disparity means a new user's question is likely to have low priority, while the experienced user will be given high priority simply for having participated over a longer period of time. A decay function applied to a user's accumulated influence would ensure that only active members are given priority over others.

PROOF. Let $I_t(q)$ be the influence of user q at time t

When $t = 0$,

$$I_t(a) \gg I_t(b)$$

Because of Rule 5, No participation by a implies

$$\Rightarrow \lim_{t \to \infty} I_t(a) = 0$$

Sustained participation by b implies

$$\Rightarrow \exists n\, I_n(b) > I_n(a))$$

As described above, this mechanism maps the reported types $\hat{\theta}$ to the desired outcome, X. Because of the revelation principle, we can say that the mechanism implements the social choice function f. Each agent fares best when truthfully reporting their type, or participating to the fullest extent of their abilities, regardless of the actions of other agents. Therefore we can say the mechanism is incentive compatible. This creates a Bayes Nash equilibrium ξ where each agent reports truthfully and earns maximal rewards. There is no incentive for agents to deviate from their strategy of truthful reporting when others have not also done so. Moreover, the strategy that arrives at this equilibrium point is a dominant strategy. Regardless of the behavior of others, it is always in the best interest of a user to answer questions to the best of his or her ability.

This mechanism has been shown to be incentive compatible. Simply put, this means the mechanism encourages beneficial behaviors in individuals, while not encouraging damaging behavior. Incentive compatibility does not mean the mechanism is optimal however. There is perhaps a better mechanism for inducing the desired outcome. The optimal mechanism is domain specific. The optimal mechanism for one QA system may not be identical to the optimal mechanism for another. Optimality can be achieved through rigorous experimentation on a specific implementation and with very precise domain knowledge. Creating an optimal incentive mechanism for QA systems is outside scope of this research.

This mechanism is incentive compatible because it implements the social choice function when operating at a Bayes Nash equilibrium point where agents participate honestly and to the best of their abilities. However, there is one potential weaknesses: collusion. Collusion occurs when multiple users work together to exploit the system. For instance, if a user with a high level of influence creates meaningless questions and a second user responds to these questions while the first user rates the answers highly, the second user will gain rewards rapidly. However, relative differences in influence are meaningful. If many users have high levels of influence, the value (question priority) for any one of those users drops. Therefore, in smaller systems the mechanism does protect against this type of fraud.

Additionally, users who are not in collusion can mark this content as spam, thereby eliminating the value to those in collusion. A more dangerous weakness is the threat of shared accounts. Multiple users operating under the same username are likely to have more expertise and availability than a single user. Therefore, it is likely that they will have a higher influence score. If many people band together under a single name each person would reap the rewards of a high influence score. Fortunately there are infrastructure-level ways to thwart this fraud. A simple example is disallowing a person to be logged in from two locations simultaneously. The following section contains an experiment that compares the performance of this mechanism to the industry standard as implemented by Yahoo! Answers.

III. MECHANISM TESTING

With these desired outcomes, *X*, and the mechanism *y* in place, it is possible to test the expected performance of this design. Ideally such tests would measure the expertise and participation levels of a population of users interacting on a live QA system. At one point on Yahoo! Answers there were approximately 120 million users and 400 million answers [18]. This yields a participation rate of roughly 3.3 responses per user over their entire lifespan on the system. The number of questions seen by each of these users, or the number of impressions, is unknown. Assume this number is 100. This means that Yahoo! Answers has a conversion rate of $0.0\overline{3}$. A 25% improvement on this performance requires a conversion rate of $0.041\overline{6}$. An *A/B* test for significance would then require > 3,000 impressions in the test group and > 3,000 impressions in the control group to show that the experimental group based on the new incentive mechanism outperforms the control group based on Yahoo! Answers with 95% significance [19]. If the measured improvement is < 25%, then more impressions would be necessary.

Such a study would involve building a fully functional QA system, recruiting several thousand users, and randomly assigning them to control and experimental groups; this type of study is outside the scope of this research. For these reasons a software simulator was created to compare an incentive mechanism based on reciprocal rewards to an incentive mechanism based on achievement rewards. This simulation was populated with agents designed to mimic human behavior in current QA sites, such as Yahoo! Answers [11]. The Python programming language was chosen to implement the simulation.

The simulation begins by instantiating a fixed number of agents *I*, each representing a human user. These agents begin with a fixed number of reward points upon instantiation. The expertise *x* of each agent *i* is represented by a normal distribution. Two fixed numbers, the expertise mean x_{μ_i} and the expertise standard deviation x_{σ_i} are unique to each agent and used to define this distribution.

$$\forall i \in I, x_i \sim N(x_{\mu_i}, x_{\sigma_i})$$

Because it has been observed that users' abilities follow a power law distribution [11], the expertise means x_{μ_i} and expertise standard deviations x_{σ_i} are assigned according the following equation, where *r* is a uniformly random number in the range [0.0, 1.0) and *m* and *s* are fixed constants. This expertise initialization matches the observed participants. There are exponentially fewer participants at the higher expertise levels.

$$\forall i \in I, x_{\mu_i} = r^m$$

$$\forall i \in I, x_{\sigma_i} = rs$$

Once these expertise models for each agent are initialized, a simulation cycle begins. One cycle is defined as a process in which:

— A random subset of the agents generates questions.

— Each agent has the opportunity to view some subset of the generated questions and estimates an expected reward.

— Each agent then ranks the questions it has seen in order of expected reward and chooses to answer a subset of these questions. This ranking is based on the expected reward calculation, which is dependent on which incentive mechanism is currently applied.

— Answers are generated and rewards are distributed based on the quality of the answer and the quality of others' answers.

— If the reciprocal incentive mechanism is in effect, then a decay factor is applied to the standing point balances for each agent.

— Some subset of the agents defect and leave the system, while some new agents are introduced.

Typically users with lower expertise are more likely to ask questions. This simulation models this as a linear relationship, where the probability of an agent asking a question in a single cycle $P(A_i)$ is defined below, where α is a constant, called the *question ask constant.*

$$\forall i \in I, P(A_i) = \frac{x_{\mu_i} + 1}{\alpha}$$

The set of all questions q_i is called *Q*. A question q_i has a difficulty, d_i, which is defined below. Note that the difficulty of the question is *not* a function of the asker's expertise, x_{μ_i}. This matches observations that experts do not necessarily ask more difficult questions. The questions may simply be in a topic that the users have very little expertise in; however, less difficult questions are much more plentiful. Therefore, this is also modeled as a power law distribution, where *r* is a uniformly random number in the range [0.0, 1.0) and *D* is a fixed constant called the *difficulty exponent*. Also, under the control mechanism based on Yahoo! Answers the agent who asks a question has 5 points deducted from its balance. There is no deduction in the experimental reciprocal mechanism.

$$\forall q_i \subset Q, d_i \quad r^D$$

Once the questions for that cycle are generated, the agents must select which questions to answer in order to maximize their reward. It is unrealistic that every agent can observe and calculate a predicted reward for every available question. This would be equivalent to a human reading the entire database of open questions on Yahoo! Answers, which numbers in the hundreds of thousands [15]. Therefore, the probability that any given question q_i is considered by agent *i*, is calculated as $P(C_{q_j,i})$.

$$|\forall i \subset I, \forall q_j \subset Q|, P(C_{q_j,i}) \quad \frac{\beta}{|Q|^K}$$

|Q| represents the number of questions, and β and *K* are constants called the *question seen constant*, and the *question exponent*, respectively. This equation indicates that as the number of questions grows, the probability of a single agent seeing one particular question shrinks exponentially. The simulator has additional functionality that can fix $P_{q,i}$= 1 for all *q* and *j*. This mode of operation emulates an ideal recommender. A recommender recommends content to users, and in the context of QA systems, it will recommend a question to a user who wishes to answer a question. An ideal recommender would examine all possible questions and return an optimal subset of questions to answer. Fixing the probability that a question is considered to 1 ensures that all possible questions are considered, and the agent can then select the questions to answer from the entire pool of questions.

Let the set of all considered questions q_j by agent *i* be called C_i. For each considered question, the agent calculates the expected reward for answering this question. This expected reward, E_{i,q_j}, is simply the probability of answering the question correctly times the reward for doing so. This reward is dependent on the incentive mechanism being used by the system. For the control group which emulates the mechanism used in Yahoo! Answers, simply supplying an answer is worth 2 points, and 10 points are given for supplying the best answer. Additional points are given for the number of times that a user "likes" the answer. See Table 1 for a full description of this mechanism [15].

TABLE I. YAHOO! ANSWERS REWARD STRUCTURE

Action	Points
Begin participating on Yahoo! Answers	One time: 100
Ask a question	-5
Choose a best answer for your question	3
No best answer was selected by voters	Points returned: 5
Answer a question	2
Deleting an answer	-2
Log into Yahoo! Answers	Once daily: 1
Vote for a best answer	1
Vote for NO best answer	0
Your answer selected as best	10
Receive a thumbs-up on a best answer	1 per, max 50
Question removed due to violation	-10

Calculating the probability of giving a best answer or the expected number of "likes" requires modeling every other agent in the system, and this is impractical for large systems. Therefore, when operating under the Yahoo! Answers mechanism the simulation agents calculate E_{i,q_j} based on the expectation of getting the answer correct, and a correct answer is worth 1 additional point. Agents know their own expertise distributions, which is $\sim N(x_{\mu_i}, x_{\sigma_i})$, and the question difficulty d_j is a fixed number between 0 and 1. Therefore, the probability of getting the correct answer equals the probability of drawing a number z_{i,q_j} from their expertise distribution that is greater than the question difficulty, d_j. It is reasonable to assume that live users are capable of determining how well they are able to answer a given question. It is much more difficult and unlikely that users will know the probability of others giving correct answers to a question.

$$|\forall i \subset I, \forall q_j \subset C_i|, E_{i,q_j} \quad 2 + P(z_{i,q_j} > d_j)$$

This expected reward E_{i,q_j} is now a function of the point total of the questioner, p_j, and a constant weight, ω.

$$|\forall i \subset I, \forall q_j \subset C_i|, E_{i,q_j} \quad P(z_{i,q_j} > d_j) p_j \omega$$

Each agent then sorts all of the considered questions by expected reward and answers them starting with the highest expected reward. The agent stops answering questions when one of three criteria occurs:

(1) All of the considered questions have been answered.

(2) The expected reward for questions q_j becomes ≤ 0.

(3) The agent has answered the maximum number of questions per cycle, a fixed constant *M*.

An answer is simply a number drawn from the expertise distribution of an agent. Rewards are calculated based on the difficulty of the question d_j, the quality of the answer, z_{i,a_j}, and depending on which incentive mechanism is used, the point total of the questioner, p_j. The answers for each question are then collected and evaluated. The best answer to a question is the answer with the highest value of those given for that question. The agent who supplied this best answer is given a bonus *B* of 10 points in the Yahoo! Answers mechanism, and a bonus B_j^* of five times the reward of an answer that is simply correct in the experimental mechanism. When the answer is not the best answer *B* = 0. These rewards R'_{ij} for the control mechanism and R_{ij} for the experimental mechanism are expressed below.

$$R_{ij} \quad \begin{cases} 3 + B : z_{i,a_j} \geq d_j \\ 2 \quad\quad : z_{i,a_j} \geq d_j \end{cases}$$

$$R'_{ij} \quad \begin{cases} p_j\omega + B_j^* : z_{i,a_j} \geq d_j \\ 0 \quad\quad\quad\quad : z_{i,a_j} \geq d_j \end{cases}$$

These points are then awarded to each user. In the Yahoo! Answers website, there is a problem of users copying content from the answers of others in an attempt to create the most comprehensive answer. This predatory behavior is called sniping. Such fraudulent behaviors are discussed further in Section 6. One of the rules of the experimental mechanism is that users cannot see others' responses until the question has been closed and rewards distributed. This rule eliminates the threat of sniping. This simulation recreates Yahoo! Answers in a favorable light because sniping is not possible. Also, note that zero points are awarded in the experimental mechanism if the correct answer is not achieved. This is done to eliminate the incentive to create worthless, or spam answers.

In the control mechanism points are accumulated, and then they are spent when asking a question (see Table 1). This discourages people from asking questions. Often those with the most expertise ask very few questions, if any. They do not want to risk their leader board standing. These are the most valuable people in the community, yet they are punished by the control incentive mechanism. In the experimental mechanism, asking questions is not discouraged, and there is no penalty for doing so. Under this mechanism, the relative difference in points accumulated has real value. Because their questions are "worth more," the leaders are given priority consideration when asking questions. In order to prevent this from becoming an exclusive club and discouraging new users from participating, new users are instantiated with a balance of 100 points, and point balances undergo a time decay in the experimental mechanism. In the simulation after each cycle the point balances are reduced by ϵ under the experimental mechanism.

A major obstacle that many online communities face is user attrition. In order to best model a real community operating in the steady-state, this simulation models the influx of new users and the defection of current users. The simulation built in models incoming new users until a certain graph density is achieved [11]. This represents the bootstrapping problem of how an online community is formed, but it does not accurately capture the steady-state operation of a mature community. To simulate defection the agents are ranked according to point balance. The probability of agent *i* defecting $P(F_i)$ is then a function of their percentile rank *t* in the system. This ensures that the most successful agents are very unlikely to defect, while those who have difficulty accumulating points are much more likely to defect. This matches observed patterns on Yahoo! Answers and other online communities [15].

$$P(F_i) \quad \frac{1}{3\sqrt{t}} \quad 0.2$$

Using this equation for calculating the probability of defection $P(F_i)$, the expected number of defectors can be calculated by solving the definite integral:

$$\int_0^1 \frac{1}{3\sqrt{t}} \quad 0.2dt \quad 0.0666$$

This means that roughly 6.7% of the all the users will defect in any given cycle, and lower ranked users are much more likely to do so. This more accurately represents the behavior in a live system than simply eliminating the lowest performers. To balance this attrition, new users are introduced. The number of new agents, or users, added every cycle is determined by an integer that is $\sim N(0.07|I|, 0.01|I|)$. Recall that $|I|$ is the number of users in the system. These two equations representing defecting existing users and the creation of new users are balanced. This is designed to model the steady state operation of a QA system with very slow growth, and these parameters can be adjusted to model other scenarios. Table II summarizes the simulation parameters used for the experiments in following section.

TABLE II. SIMULATION PARAMETERS

Name	Symbol	Value
Initial number of users	$\|I\|$	250
Expertise exponent	m	3
Expertise standard deviation multiplier	s	0.2
Question ask constant	α	10
Difficulty exponent	D	3
Question seen constant	β	2
Question exponent	K	0.5
Reciprocal reward weight	ω	0.1
Maximum number of answers	M	5
Point decay percentage	ϵ	5%

IV. TESTING RESULTS

The simulator was run in several different configurations for fifty complete cycles. One set of fifty cycles completes a single round. Results were then collected after twenty rounds have been completed. Reward points and users persist between cycles, but there is no concept of state that is preserved between rounds. Because we are most concerned with the behaviors of experts in this simulation, this section analyzes the performance of the top performing agents. Reward points are an artificial construct designed to encourage participation among the experts. Analyzing point accumulation alone is not meaningful; therefore, the analysis presented here focuses on expertise and participation, which have a measurable impact on the usefulness of the community. Table 3 contains the data collected from the top 10% of the point earners after fifty cycles, averaged over twenty rounds.

TABLE III. TOP 10% EARNERS

	standard		reciprocal		reciprocal*	
Measure	$\bar{x}$	s	$\bar{x}$	s	$\bar{x}$	s
Expertise mean, μ_i	0.745	0.151	0.838	0.118	0.886	0.0745
# Questions asked	0.941	0.957	0.569	0.869	0.462	0.776
# Answers received	66.75	42.25	62.46	52.29	83.41	83.23
# Questions answered	248.68	1.425	175.11	64.78	192.54	56.65

Table 3 shows the performance of the top point earners. In other words, these measurements characterize the agents in the system that earned the highest rewards. This can be used to evaluate the incentive mechanism because this table describes the behaviors of most rewarded agents.

The first configuration in Table 3, the "standard" column group, represents the performance characteristics of a generalized version of the Yahoo! Answers mechanism. This simulated mechanism is actually expected to perform better than the authentic version because the simulated version is immune to sniping and spam. The second column group contains the performance measurements of the reciprocal mechanism. The final column group, labeled "reciprocal*", duplicates the functionality of the "reciprocal" mechanism, but with the added component of an ideal recommender. Essentially this recommender mimics the functionality of an omniscient recommender because it allows agents to evaluate all questions and pick the most suitable ones to answer. Ordinarily an agent has a limited pool of questions under consideration, which models the human usage of a QA system. On a system of any appreciable size, no user has the ability or inclination to read every question.

Because each of the measurements in Table 3 represents data that are collected from twenty rounds of fifty cycles each, the measurements are expressed as a sample mean, $\bar{x}$ and a sample standard deviation s. The size of each sample is then equal to the number of rounds, in this case twenty.

The first row contains the average expertise mean, μ_i, for the agents in the top 10% of point earners. Recall that agents are instantiated with an expertise mean drawn from a power law, and it is bounded between zero and one, with lower values much more likely. Both the reciprocal and the reciprocal* mechanisms are more effective than the standard at rewarding the agents with the highest expertise. This difference is statistically significant according to an independent, two sample, two tailed *t*-test for statistical significance ($\alpha < 0.05$). Fig. 1 illustrates this.

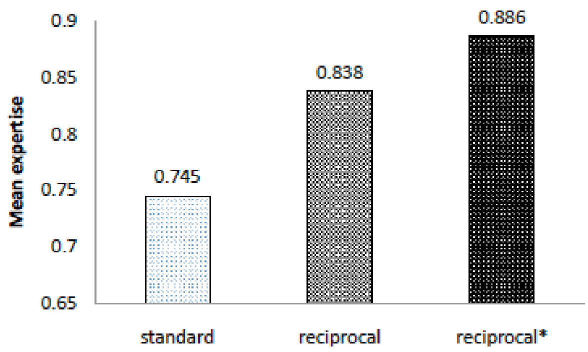

Fig. 1. Mean expertise of the top 10% of point earners.

The next row shows the average number of questions asked by the top ranked agents. As expected, the top ranked agents do not ask many questions. This is because the probability of asking a question is inversely proportional the expertise level of the agent, and naturally the top earning agents are those with the most expertise. Due to the high sample standard deviation values the differences within this row are statistically significant only with a value of $\alpha < 0.02$.

The following row contains the average number of answers received per top ranked agent. Here again the reciprocal mechanisms perform at least as well as the standard mechanism. Consider that under the reciprocal mechanisms the agents asked roughly half as many questions as the agents adhering to the standard mechanism, yet they receive as many or more (in the case of reciprocal*) responses. Therefore, under the reciprocal mechanism the top performers receive twice as many responses for their questions. This is a key strength of the reciprocal mechanism. These extra responses comprise the systemic reciprocal reward.

The final row in Table 3 shows the number of questions answered by the top-earning agents. The standard mechanism consistently yields a significantly larger number of questions answered than the other mechanisms, as indicated by the highest mean and small standard deviation. This is because the Yahoo! Answers mechanism rewards simply providing an answer, as indicated in Table 1, regardless of correctness or if it comes from a reliable, expert source. Additionally, it is suspected that this is due to the bootstrapping dynamics of the experimental mechanisms. Under these reciprocal mechanisms, each agent starts on a level playing field, but it is possible to gain rewards more rapidly than in the standard model, causing fragmentation within the community. An agent with high expertise which fails several questions shortly after instantiation will be much more likely to defect from the system, as described previously. Ultimately, this means that some of the highest achievers are relative newcomers to the community, and they simply have not had the time to answer as many questions. Under the standard mechanism those with the most expertise slowly percolate to the top, and they tend to stay there for a long time and answer many questions.

Table 4 is very similar to Table 3, but instead of measuring the agents from the top 10% of point earners, it contains measurements from the agents ranked in the top 10% according to expertise, [illegible]. Previously, Table 3 can be used to evaluate the incentive mechanism directly. Table 4 is perhaps even more interesting because it shows how the top experts in the system behave.

TABLE IV. TOP 10% OF EARNERS BY [illegible]

	standard		reciprocal		reciprocal*	
Measure	$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
Expertise mean, [illegible]	0.896	0.0585	0.941	0.0339	0.941	0.0342
# Questions asked	0.321	0.603	0.153	0.380	0.162	0.404
# Answers received	23.34	27.04	16.57	23.15	28.40	45.30
# Questions answered	146.55	98.56	133.8 7	86.07	142.31	85.99

The first thing to notice is the strong similarity between these two tables. The expertise means in the first row of Table 3 are close to the true maximum expertise means as shown in Table 4. This assures us that all of the tested mechanisms are fairly effective at identifying the experts and that the reciprocal mechanisms outperform the standard mechanism. All mechanisms instantiate users in the same manner. The only remaining explanation for the discrepancy in expertise mean values ([illegible]) between the different mechanisms is that the standard mechanism kept some of the best experts at an artificially low reward level and promoted lesser agents, thus increasing the chance of expert defection. Also notice in this table the difference in the number of answers received per question asked across the three mechanisms is even more apparent (illustrated in Fig. 2.)

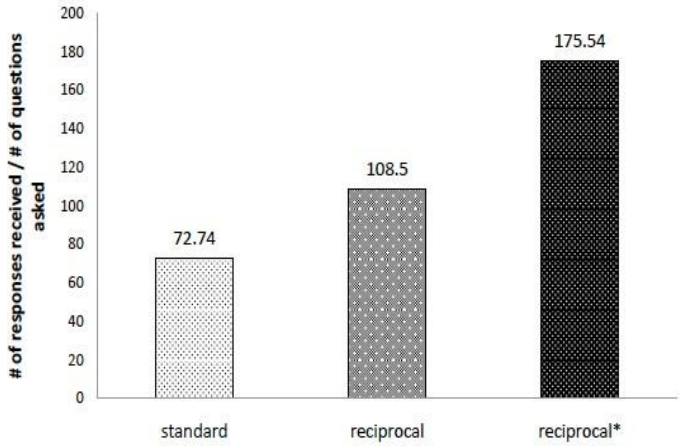

Fig. 2. Mean expertise of the top 10% of point earners.

The final row in Table 4 shows that the agents with the highest expertise do not answer significantly more questions under the standard mechanism. Therefore, the reciprocal mechanisms are just as good as the standard one for encouraging experts to answer questions. This section demonstrates that in a software simulation the reciprocal mechanism outperforms the standard Yahoo! Answers based mechanism according to several different metrics, including rewarding expertise and drawing a larger number of responses per question asked by experts (shown in Fig. 1 and 2).

V. FRAUD

Because the reward of system influence has real value, users may be tempted to cheat the system to unfairly collect larger rewards. This hurts others in the system because the relative disparity in influence scores helps to drive others to participate. Thus, artificially inflated scores can unbalance the system.

Fraud can have many different forms. Often in aggregative systems a user may simply create spam for one of two reasons. First is to advertise an outside product or service in which the user has an interest. Thankfully we can rely on the collective wisdom of the other users to identify this behavior and nullify its effects. Another type of spam is the contribution of relevant but low quality content in an attempt to inflate one's participation score. From a system perspective this appears almost identical to a situation where a user with little expertise but much free time contributes much content at the best of his ability—a situation we wish to encourage. Fortunately this type of behavior is not rewarded as greatly as the case when a user demonstrates true expertise. This is due to the collaborative nature of the incentive mechanism. Endorsements from users with high expertise would allow greater influence achievement when contributing valuable content.

The incentive mechanism must be designed to combat various types of fraud. Some obvious fraud deterrents include harsh penalties for getting caught performing fraudulent behavior. Perhaps less obtrusive, this research combats fraud through the principle of incentive compatibility. A mechanism is said to be incentive compatible if every participant fares best (earns the most reward) by truthfully sharing private information, or in the context of QA, participating to the best of his ability. Though widely adopted, the mechanism behind Yahoo! Answers has a design flaw: it provides an incentive for answer *sniping*. Sniping occurs when a responder may not know the answer to a question and simply collects pertinent pieces of the answers from previous responders. Then this responder may have the most comprehensive answer, though he/she did not add any new content.

This act of assembling the information does have some value, but rewarding this user instead of the original contributors does them a disservice. This mechanism is not incentive compatible with a desired outcome because users have an incentive to snipe answers [20]. They suggest a rule where the asker distributes the rewards across multiple answer contributors. The mechanism in this research has a more elegant solution to this problem: responders cannot see the responses of others until the question answering period is completed, as decided by either by the original questioner or a system-wide policy based on activity and time. This simple change to the mechanics of a QA system eliminates the threat of answer sniping while maintaining author integrity.

Several types of fraud may be more difficult to prevent using incentive mechanism design. One consideration is that users may simply create positive feedback for themselves. Because the influence score is based on the feedback and influence of the rating user, a user who evaluates his content positively would create an infinite loop. This is thwarted in the proposed QA system by simply disallowing a user from rating his own content. This type of self-feedback can also be created in a more sophisticated manner by collusive voting. Either multiple users may set up secret agreements to provide positive feedback, or a single user may have multiple accounts which evaluate each other. Trust can be applied at the application level to detect this type of fraud. Users who provide ratings that are not corroborated by others may be identified as fraudulent. If it can be observed that two or more users regularly give each other positive feedback in the absence of positive feedback from others, it is possible that this type of fraud has occurred. Such sophisticated techniques for detecting and mitigating fraud are outside the scope of this dissertation.

Perhaps an even more difficult type of fraud to detect is shared accounts. From a system perspective, multiple users on the same account would appear as a single user with very high participation and a broad body of expertise. This could enable rapid growth of influence, and each person could reap the benefit of asking questions under this username. It is likely that many questions of very diverse topics and length would originate from this single user account. This could possibly be detected by analyzing the user's question to answer ratio or the question topic diversity. On the network level this type of fraud may become more apparent by analyzing the IP address of the content origin. A shared account would likely have simultaneous people logged into the account from many different IP addresses.

Fraud has the capability to cripple an online community. It is imperative to combat this fraud with all necessary means while maintaining system functionality. Application level solutions, like trust, can be combined with network level solutions, like traffic and IP monitoring, to detect fraudulent behavior. The most important step toward fighting fraud is to remove any incentive to perform fraudulent behavior through careful mechanism design.

VI. Conclusions

The purpose of this research is to discover how to encourage expert participation in online communities. These communities are growing rapidly and we have come to rely on them as a source of valuable information and entertainment. They can take many forms, including a question and answer system, a news aggregation service, a discussion forum, or a social network, just to name a few. Most of the research presented here pertains to QA systems as an example, but it is adaptable to other forms on online communities as well. Experts across these various communities are those who add the most value to the community, therefore their participation is highly desired.

A promising area for future work involves examining the behavior of live users operating under the novel incentive mechanism. This work shows that live survey respondents have expressed preference for such a mechanism, but this is not necessarily an indicator of how they would behave in a live system. A test in a real-world environment would make a more convincing argument for a new class of incentives. Additionally, research on a live system would allow further development toward an optimal mechanism. The parameters of the simulation, while based on observations and analysis of existing QA forums, have been chosen to best approximate generalized behavior in QA systems. Mechanism optimization is highly domain dependent, and different QA systems have very different dynamic behavior.

A natural continuation of this work involves adapting this class of incentive mechanisms to other types of online communities. An adaptation to networking-oriented social networks such as Facebook and LinkedIn would be particularly interesting. Currently these sites rely primarily on intrinsic rewards; linking to someone is its own reward. A layer of incentives on top of such communities could spur future growth.

Overall, this research proposes an engineering solution to a fundamentally human problem. The proposed expertise modeling process, recommendation architecture, and incentive mechanism are designed to lower the barrier to entry and encourage expert participation in online communities. Increased expert participation ensures added value, and in the context of QA systems, accurate solutions and satisfied questioners. The impact of this work extends beyond QA and applies to peer production systems in general. The research presented here is the first to show how content generated by peers, with no intermediate monetary value, can directly motivate people to apply their expertise and effort toward a socially beneficial system.

ACKNOWLEDGMENT

The authors would like to thank The Center for Identity at the University of Texas at Austin for supporting this research.

REFERENCES

[1] Arpita Ghosh and Preston McAfee, "Incentivizing high-quality user-generated content," in WWW '11 Proceedings of the 20th International Conference on World Wide Web, 2011, pp. 137-146.

[2] N. Thurman, "Forums for citizen journalists? Adoption of user generated content initiatives by online news media. *New Media & Society*, vol. 10(1), p. 139, 2008.

[3] Arpita Ghosh, "Social computing and user-generated content: a game theoretic approach," in ACM SIGecom Exchange, vol. 11, pp-16-21, December 2012.

[4] Al M. Rashid, Kimberly Ling, Regina D. Tassone, Paul Resnick, Robert Kraut, and John Reidl, "Motivating participation by displaying the value of contribution," in CHI '06 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2006, pp. 955-958.

[5] D.C. Parkes, Iterative combinatorial auctions: Achieving economic and computational efficiency, PhD thesis, Citeseer, 2001.

[6] Tuomas Sandholm, "Automated mechanism design: A new application area for search algorithms," Lecture notes in computer science, pp. 19-36, 2003.

[7] Suratna Budalakoti, David DeAngelis, and K. Suzanne Barber, "Expertise modeling and recommendation in online question and answer forums," in SIN '09: Proceedings of the IEEE Conference on Social Computing, Symposium on Social Intelligence and Networking, Vancouver, BC, Canada.

[8] Suratna Budalakoti, David DeAngelis, and K Suzanne Barber, "Unbiased trust estimation in content-oriented social networks," in Proceedings of The Workshop on Trust in Agent Societies at The Tenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2011), 2011, Taipei, Taiwan.

[9] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth, "The author-topic model for authors and documents," in AUAI '04: Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence, 2004, pp. 487-494, Arlington, Virginia, United States. AUAI Press.

[10] Jun Zhang, Mark S. Ackerman, Lada Adamic, and Kevin Kyung Nam, "A mechanism to support expertise standing in online help-seeking communities," in UIST '07: Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology, 2007, pp. 111-114.

[11] Jun Zhang, Mark S. Ackerman, and Lada Adamic, "Expertise networks in online communities: structure and algorithms," in WWW '07: Proceedings of the 16th International Conference on World Wide Web, 2007, pp. 221-230.

[12] Pawel Jurczyk and Eugene Agichtein, "Discovering authorities in question answer communities by using link analysis," in CIKM '07: Proceedings of the 16th ACM Conference on Information and Knowledge Management, 2007, pp. 919-922.

[13] Mohamed Bouguessa, Benoit Dumoulin, and Shengrui Wang, "Identifying authoritative actors in question-answering forums: the case of Yahoo! Answers," in KDD '08 Proceedings of the 14th ACMSIGKDD International Conference on Knowledge Discovery and Data Mining, 2008, pp. 866-874.

[14] Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne, "Finding high-quality content in social media," in WSDM '08 Proceedings of the 2008 International Conference on Web Search and Data Mining, 2008, pp. 183-194.

[15] Yahoo! Answers Team, "Yahoo! Answers blog: Did you know?" November 2009, Retrieved from http://yanswersblog.com/index.php/archives/2009/10/05/did-you-know/.

[16] Leonid Hurwicz and Stanley Reiter, Designing Economic Mechanisms, Cambridge University Press, 2008.

[17] R.B. Myerson, "Optimal auction design," Mathematics of Operations Research, vol. 6(1), p. 58, 1981.

[18] Jacob Leibenluft, "A Librarian's Worst Nightmare: Yahoo! Answers, where 120 million users can be wrong," Slate Magazine, 2007.

[19] J.A. Mazanec and H. Strasser. A nonparametric approach to perceptions-based market segmentation: Foundations, volume 1, Springer Verlag, 2000.

[20] Shaili Jain, Yiling Chen, and David C. Parkes. "Designing incentives for online question and answer forums," in *EC '09: Proceedings of the tenth ACM conference on Electronic commerce*, 2009, pp. 129-138.