Trust-Based Community Formation in Peer-to-Peer File Sharing Networks

Yao Wang, Julita Vassileva
Department of Computer Science
University of Saskatchewan, Saskatoon, SK, Canada
{yaw181, jiv} @ cs.usask.ca

Abstract

Decentralized peer-to-peer (P2P) networks can benefit from forming interest-based communities that can provide peers with information about the resources shared in the community and collectively computed rating of their quality as well as about the agents in the community and their reputation. We propose a mechanism for forming communities in a P2P system for sharing academic papers. The mechanism requires each agent to compute its trust in the agents with whom it interacts. A simulation shows that such communities can benefit peers.

1. Introduction

A P2P file sharing system for research papers called Comtella [3] has been developed at the MADMUC Lab in the University of Saskatchewan. It allows users to share not only the papers that they are interested in (both as files or as links/ bookmarks), but also their evaluations and comments about the papers. When users search for papers that other users share in Comtella, they can also see other users' ratings and comments, which helps them to decide whether a paper is worth to read and to compare their opinions.

Although researchers can benefit from a system like Comtella, there is a serious usability problem of information overload. Typically the result of a search is a long list of papers. The ratings of the papers made by other users can give indication of the paper's quality, but people rate papers differently since they have different interests and knowledge. Therefore it is hard to decide which papers are good without knowing anything about the user who gave the rating. Agents can be used to keep track of similarity in the ratings given by their users. The agents representing users with similar interests and knowledge level can form trusted communities that can recommend appropriate papers to the community members.

The rest of this paper is organized as follows: section 2 discusses related work in the area of agent group-formation. Section 3 presents our approach to forming communities. The experiment design and results are described in section 4. Section 5 discusses the results, related work and directions for future work. The last section presents the conclusions.

2. Groups of agents and agent communities

Depending on the closeness of cooperation, the duration and commonality among agents, three main types of agent groups have been proposed in the Multi-Agent Systems (MAS) community: teams, coalitions, and congregations.

Teams of agents are composed of agents cooperating to solve a problem that cannot be solved by any individual member or to solve a problem more efficiently. Examples of such groups are robot-soccer and robot-rescue teams. The main research focus in this area is the assignment of tasks and the coordination among the agents [8, 11].

Coalitions of agents are mostly used in e-commerce and composed of self-interested agents. The motivation for agents to join a coalition is that they can get more benefit as members of the coalition, although they can act alone. Coalitions are typically short-term groups. Research on coalitions focuses on how to form coalitions that maximize the individual and group utility and how to distribute the utility among the members [7, 12].

Congregations [2] are long-term groupings of self-interested agents, formed in a self-organized way by dividing a large group of agents into subgroups composed of agents with the same interests. The benefit of forming congregations is that agents can more easily find each other and have more successful interactions with agents in the same congregation. Agents in congregations are more loosely coupled than the agents in coalitions or teams.

We define an agent *community* as an organization that facilitates a group of agents sharing common interests and preferences to share knowledge, learn and benefit from one another. A community of agents has similarities with all types of agent groups described above. In a community the agents work together to achieve some common goals, which can facilitate achieving their individual goals. The goals of a community are long-term goals and are hard to express in terms of individual utility, unlike the goals of agents who enter a coalition. For example, in a P2P network, a community can serve as an information center to provide agents with integrated information that would otherwise be distributed in each peer. A community is similar to a congregation since it brings like-minded agents together and helps them find each other and have more successful interactions. Like a team, a community is organized; some agents can take specific community-related roles, which is not the case in congregations. In contrast with coalitions and congregations where agents interact only within their group or with other agents as a group, the agents who are members of a community are free to interact with non-members. This is beneficial for the community (allows access to resources outside of the community and locating potential new members) and for non-members of the community who can use some of the community services to access resources of the community.

3. Community-formation mechanism

In Comtella, each shared paper is associated with a category (or subject area) specified by the paper provider (the user who first shares the paper). Currently all the users use the same categories to annotate and search for papers, so the categories serve as ontology for the research areas represented by the papers in the system. Users search for papers by categories. Users can also search for communities in a given category. From the communities, they can find who shares papers in the area described by the category and who are the most reputable users (peers) in the community so that they can pay more attention to the papers or ratings provided by them. Users can also learn which papers are good according to the collective ratings from the community. In order to build such communities, several issues have to be addressed.

3.1 Issues and mechanism outline

There are a number of issues to be considered when designing a community formation mechanism.

1. Who will create a community? Any agent can create a community as long as its user wants to contribute more resources, such as computer processing time, disk space, and bandwidth, to service the community. We call such an agent creator.
2. Who will be eligible to be a member of the community? It is desirable that the members are agents whose owners share good papers and provide competent and fair ratings.
3. How does an agent decide to join or leave a community?
4. What are the responsibilities of the community members?
5. How does a community evolve? A community can be created, destroyed or annexed.
6. Can an agent join multiple communities at a time?

We assume that an agent can join only one community for one particular category of interest. However, since the user can have interest in many different categories of papers, his/her agent can join multiple communities for different categories. A community integrates information collected from the individual members and provides it to its members and to other agents who are looking for papers in the category of the community.

Since a creator needs to dedicate some resources to build a community, it would prefer to build a community that is useful, for example in a subject area of strong interest for its user. In order to find out which agents provide good papers and ratings, the creator needs to build trust in other agents by learning from its experience with these agents. Once the creator finds trustworthy agents, it invites the agents to join its community. If an invited agent judges the community as being trustworthy and if it has not joined another community in the same category of interest, it will join the community. Once an agent joins a community, it invites its own set of trustworthy agents in the category to join the community. In this way, the agents in the community help each other to find other potential trustworthy agents and the community grows quickly. An agent can join a community only when it is invited by one of the current community members. An agent is free to leave a community, if the community becomes no longer trustworthy from the agent's point of view.

As already mentioned, there can be multiple communities for a particular category. When an agent is a member of one community, but is invited by a member of another community to join the second community, it can suggest to the two communities to join. If they refuse, it will join the more trustworthy community according to its own judgement criteria. Suggestions for joining communities are considered by all members of both communities. If most of the members in each community judge the other community as trustworthy, the two communities will join to form a new, bigger community with more resources and therefore with a bigger capacity. Such a community can provide more resources and better ratings [5].

Each community can only have a limited number of members, called *community capacity* and computed as a function of the number of the creators and their community-dedicated resources. The users who wish to be creators can decide how many community members they want to support and this will define what amount of their disk space and CPU can be dedicated to serving the community. When a new creator joins a community, the resulting community capability will increase as a function of the number of creators and the resources they contribute.

The next sections will discuss the metrics for individual trust in other agents, collective trust in a community and for community capacity, used in the community-building mechanism.

3.2 Trust in an agent

We define the notion *"trust"* as a measure used by people to evaluate (based on their expectations and using their previous experience) other people's capability of providing a good quality service or resource and their capability to judge the quality of service or resource truthfully. This definition of trust combines elements of two other definitions, [14] which emphasizes that the performance of the trusted person/agent has to meet or exceed the expectations of the trusting agent, and [15], which emphasizes similarity in judgment criteria between the trusting and the trusted person/agent. Other definitions have been proposed, which emphasize different aspects of trust, for example [4] which emphasizes the truthfulness / sincerity in communication between the trusting and trusted person/agent. Our definition doesn't cover this aspect, since it is unlikely that users in a system like Comtella will try to misrepresent their ratings or be dishonest. A basic assumption in our definition, as in all other trust-based systems is that agents / users are uniquely identifiable (whether by real name or alias).

In Comtella, user A will trust user B, if A's experience shows that B has provided in the past similar ratings to its own rating of the paper, i.e. ratings that correspond to A's understanding of the domain and quality criteria. Agents represent users and maintain trust representations in each other according to their users' paper ratings. Since the papers are classified into categories, and users can have different competence and quality criteria in different categories of interest, agents build trust in each other for each category.

Each downloaded paper is associated with a *provider* and an *original provider*. The provider is the user from whom another user downloads the paper. The original provider is the user who originally introduces the paper into the system. Let's say user A shares a paper that he finds on the web with Google. User B downloads the paper from A. Then user C downloads the same paper from user B. For user B, the provider and original provider of the paper are the same user, A. For user C, the paper's provider is B, but the paper's original provider is A.

The agent updates its trust in the paper provider when its user rates the downloaded paper. Users can choose to share papers without giving ratings and comments. So there are two situations when the person's agent updates its trust in the provider of the paper: when the paper provider has rated the paper and when the paper provider has not rated the paper. The agent will build two components of trust according to the two situations. The first one is the trust in the paper provider's capability of providing good ratings, if the paper provider provides a rating. The second one is the trust in the paper provider's capability of providing good papers, if he has no rating for the paper. An agent's overall trust in a paper provider is the combination of the two components. We assume that users rate the papers using a 5 scale rating scheme, from "0" (worst) to "4" (best).

Suppose the user's rating is *u*. If the paper provider gives a rating *e*, the user's agent will compare the two ratings and update its trust in the paper provider's capability of providing good ratings, *trust_rating*, by Formula (1).

$$trust\_rating = \frac{m}{n+a} \quad (1)$$

where *m* and *n* are the number of the similar ratings and the total number of the ratings given by the paper provider. *a* is a constant used to adjust the value of *trust_rating*.

Given a rating-difference tolerance threshold *t*, if $|u - e| < t$, the paper provider's rating is considered similar. The value of *t* can be 1 or 2. When $t = 1$, it means that the paper provider's rating is considered similar only when *u* is equal to *e*. When $t = 2$, it means that the paper provider's rating is considered similar when *u* is equal to *e*, *e -1* or *e +1*. When the value of *n* is small, no matter what the value of *m* is, the value of *trust_rating* should be smaller than 1. When the value of *n* is big enough, the value of *trust_rating* is almost equal to the proportion of similar ratings. The purpose of *a* is to make *trust_rating* smaller when the agent has not accumulated enough experiences with the paper provider. This trust component represents the definition of trust that agents trust agents with similar tastes and judgement criteria.

$$trust\_noRating = \frac{m'}{n'+a} \quad (2)$$

Formula (2) is used by an agent to measure its trust in the paper provider, *trust_noRating*, when the paper provider has not rated the paper. Given a rating-threshold *t'*, if the user's rating is equal or higher than *t'*, we say the interaction is successful. *m'* and *n'* are the number of successful interactions and the total number of interactions when the paper provider's ratings are unavailable. This trust component corresponds to the definition of trust, that the service / good provided has to meet or exceed the expectations of the trusting party [14].

$$trust = r * \frac{n}{n+n'} * trust\_rating + \frac{n'}{n+n'} * trust\_noRating \quad (3)$$

Formula (3) shows how the overall trust, *trust*, is computed. The two kinds of trust, *trust_rating* and *trust_noRating*, have different weights, denoted by $\frac{n}{n+n'}$ and $\frac{n'}{n+n'}$ respectively, which are the fractions of the total number of interactions. *r* is a value bigger than 1, used to increase the weight of *trust_rating* to ensure that users who give more competent ratings can gain trust more quickly than users who do not give ratings. *Trust* is used to measure whether an agent is trustworthy. Given a threshold *tt*, if *trust* > *tt*, the agent is considered trustworthy.

After updating the trust in the provider of the paper, the agent of the user will update its trust in the paper's original provider using Formula 2. The rating given by the original provider is not considered here, since we do not assume that it is downloaded with the paper (only the rating from the provider will be available typically). Given two paper-rating thresholds, *v1*, *v2*, where *v1* <= *v2*, if *u* < *v1*, the user rates the paper as bad and his/her agent will reduce the trust in the original provider by increasing the value of *n'* by 1. If *u* > *v2*, the user rates the paper as good and his agent will increase the trust by increasing the values of *n'* and *m'*. If *v1*<= *u* <=*v2*, the trust in the original provider will not be updated.

In the system, the original providers are important since they introduce new papers which will be propagated to other users. If the papers are good, they will benefit more users, but if they are bad, more users will waste time and effort to download and rate the papers. So whether an original provider can introduce good papers is essential for the system. If the original provider introduces a good paper, his/her reputation, a collective evaluation of the provider's capability of providing good papers and rating, will increase quickly, since no matter whether the other users download the paper directly or not, they will all increase their trust in the original provider. If the original provider introduces a bad paper, his reputation will drop quickly for the same reason. The reputation updating mechanism is discussed in the next section.

3.3 Collective trust measures

There are two collective trust measures: an agent's trust in a community as a whole and an agent's reputation among the members of a community. An agent' trust in a community is based on its trust in the individual members of the community (4):

$$trust\_community = \sum_{q=1}^{r} trust_q \qquad (4)$$

where *trust-community* denotes the agent's trust in the community. *r* is the number of the community members that are trustworthy. $trust_q$ is the agent's trust in the *qth* trustworthy community member. Given a threshold *tc*, if *trust_community* > *tc*, the community is trustworthy. Otherwise, it is not trustworthy. It is clear from the formula, that the trust in the community is proportional to the size of the community and to the trust in each member of the community. This measure is not normalized, i.e. it grows proportionally to the size of the community. This is deliberate; the size of the community matters, since more peers are likely to share more papers, so if an agent is searching for an article, it is more likely to find it in a larger community. A small community of very trusted agents is less likely to be useful for searching papers than a larger community with less trusted agents, a phenomenon similar to "the strength of weak ties" shown in studies with real communities [6].

The agent's reputation within a community is a collective measure of how much an agent is trusted by the members of the community. It is computed as follows (5):

$$R_i = \frac{\sum_{j=1}^{l} trust_{ji}}{l+b} \qquad (5)$$

where R_i denotes the reputation of the *agent i*. *l* is the number of the community members that trust the agent *i* more than their community recommendation threshold (see next section). $trust_{ji}$ is the trust of the *jth* recommending member in the *agent i*. *b* is a constant, similar to *a* in Formula (1). If an agent is trusted by more community members, it will have a better reputation. If the community members trust the agent more, the agent will also have a better reputation.

3.4 Community update

The community maintains collective information that is served to the community members and other agents who contact the community. This information includes three lists that characterize a community: a creator list, a member list, and a paper list. These lists are part of the community service (directory) and are maintained by the creators. The creator list stores the information about the creators, including the creator IDs, IP addresses, reputations, and information about their shared resources. The member list includes similar information about the community members. The paper list is used to store information about the papers shared by the members and creators, such as paper titles, links, and ratings. When multiple creators work in the same community, they share their resources and organize themselves to provide a centralized view to all community information. Each creator can store all or part of the community information, which depends on its dedicated resources. If a creator has not enough resources to keep all the information, it will provide links to other creators that keep the rest of the information. Other agents can get the information from a community by connecting to one of the creators. If there are several creators the community information is stored redundantly which enhances the availability and reliability of the community information in case one of the creators is unavailable or leaves the community. The creators in our mechanism bear some similarity to the super peers in super-peer networks, like Kazaa or Gnutella v6, where a group of super peers work together to serve a set of peers and collect and maintain information from these peers. Therefore mechanisms such as [16] proposed to improve the robustness of the super peer group can also be applied to the communities in our mechanism.

In our mechanism, a community updates its member list and paper list periodically since any member can introduce new agents into the community without the consensus of other members. This straightforward way to introduce a new member into the community makes it easy also to bring bad agents into the community by mistake. Also agents may change their behaviour due to change in their interests. Therefore a community needs a mechanism to find these bad agents and expulse them.

An agent's *reputation* within a community can be used to judge whether the agent is good or bad. Each member recommends to the community their trustworthy agents. Then the community computes the reputations of the agents recommended by the community members.

The community sorts all the recommended agents by their reputations (this list includes all the members of the community, if they are still trustworthy enough to be among the recommended agents) and asks the top agents with the best reputations to join, if they are not already in the community. The community members that are neither the creators nor the top recommended agents will be expulsed from the community. This is how a community purges itself from members who have become less trusted. A creator can not be expulsed from its community even when its reputation is below the given threshold since the creator contributes resources that the community needs. In that case, the creator will not be asked for evaluating other agents or papers. A creator can leave a community when the community is no more trustworthy from the creator's point of view. When a creator leaves a community, if the number of the community members exceeds the community capacity, the community members whose reputation values are low will be expulsed.

After updating the members, the community will update its paper list. The community will collect information from its members about their shared papers and ratings. If a paper has been rated by more than one member, the community will give a collective rating of the paper (6).

$$rating\_community = \frac{\sum_{z=1}^{g} rating_z}{g+1} \qquad (6)$$

where *rating_community* denotes the collective rating of a paper by the community. $rating_z$ is the *z-th* rating of the paper from the community members. *g* is the number of the ratings of the paper.

4. Simulation design and results

To evaluate our approach, we developed a simulation of the P2P file sharing system using JADE 2.5. Since an agent can only join one community for one category, for simplicity, we used one category in the simulation. However, there can be more than one community in this category.

Our experiments involve 50 agents where 50% of the agents are creators. A creator can contribute resources to support 5 members. Therefore a community's capacity can be defined as 5**n* where *n* is the number of creators in the community. Each paper has an intrinsic quality, *qua*, represented by a value between 0 and 1. Each agent initially shares 10 papers with different qualities. We use three kinds of agents to model three kinds of users depending on their expertise. Each agent is associated with a value, *dev,* indicating the extent of their knowledge – the smaller the value *dev*, the more expertise the agent has. An agent rates a paper according to the value generated from Gaussian distribution with the mean equal to *qua*, and deviation equal to *dev.* The value is truncated in a range [0, 1], which is segmented into 5 sub-ranges. Each sub-range corresponds to a rating from 0 to 4. The smaller the value of *dev*, the closer the rating of the agent will be to the intrinsic quality of the paper.

An interaction happens when an agent sends a request for a paper. The agent has the choice to request the paper from the community to which it belongs (if such community exists), from other communities, or send a general query to other neighbouring peers, in the way queries are sent in Gnutella. In our experiments, agents will search papers in their own communities with a 30 % chance, in other communities with a 20% chance and from the general population with a 50% chance. If the agent is searching for a paper in a community, it will choose the most highly recommended paper by the community. When selecting papers from outside a community, it will choose the papers with the best rating *mv* according to formula (7).

$$mv = trust\_rating * rating \qquad (7)$$

where *trust_rating* is the agent's trust in the paper provider's rating capability. *rating* is the rating of the paper from the provider. Given a threshold *tr*, if $mv < tr$, which means the agent can not find a good paper from the rated papers, the agent will randomly choose an unrated paper from the most trusted agent according to the value of *trust_noRating*. Agents will also update their neighbours (the peers to whom they send queries when they search in the entire population) by choosing their most trusted agents.

The goal of the first experiment is to see if forming communities helps agents find good papers. We have two configurations. In configuration 1, the original provider does not rate the shared papers at the beginning of the interactions. In configuration 2, 50% of the shared papers are rated by the original providers. We introduce these two configurations to test the impact on the performance of initial information (ratings) of papers in the system. In both of these configurations, agents rate all papers that they download. If an agent downloads a paper and is fairly satisfied with the quality of the paper by giving a rating over 2, we call this a *successful interaction*. If the paper is selected from a community, we call this an *interaction with communities*. Otherwise, we call it an *interaction outside communities*. We run each configuration for 10 times and compute the average ratio of the number of successful interactions vs. all interactions.

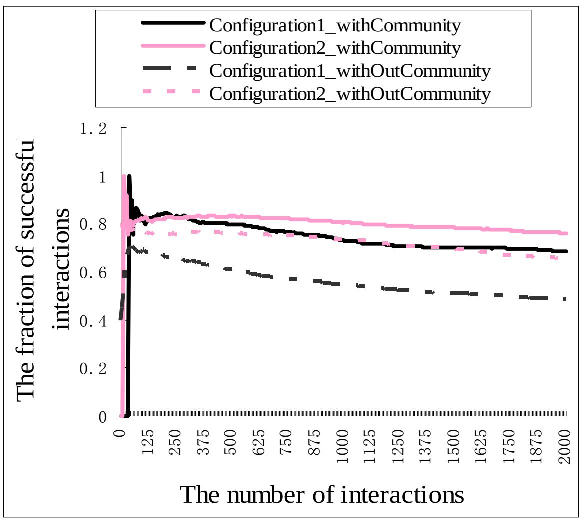

Figure 1. The performances of the systems with and without communities

Figure 1 shows the results for both configurations for interactions with and without communities. We can see that both configurations perform better in interactions with community than in interactions without community. Comparing the performance of configuration pairs with the same type of interactions (with community versus without community), we can see that the system with configuration 2 performs better than the system with configuration 1. This is expected; the starting condition in configuration 2 is more favourable for all agents, since there are initial ratings. Interestingly, the performance of configuration 1 in interactions with community comes very close to the performance of configuration 2 without community. This means that the community compensates the lack of initial ratings. The worst results are for the case when there are no initial ratings and no community – these results are about 20% worse than in the case of configuration 1 (no initial ratings) with community. We can conclude that forming communities is very beneficial for agents in finding good papers especially in a system with fewer or no initial ratings.

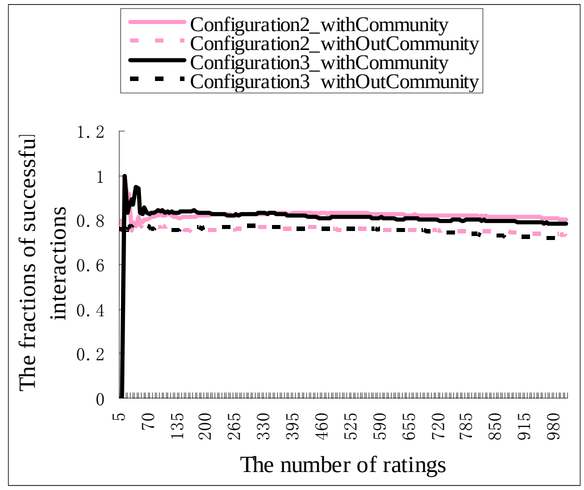

Figure 2. The influence of the frequency of agents' ratings

In the previous two configurations, agents rate each paper they download. But in a real system, people may not rate each paper. The goal of the second experiment is to see whether the frequency of agents' ratings of papers will influence the performance. We define a configuration 3, which is the same as configuration 2 (50% of the papers are rated at the beginning by the original providers), but while in configuration 2 agents rate every downloaded paper, in configuration 3 the chance that they rate it is only 50%. We compared the performance of configurations 2 and 3. Figure 2 shows the results using the number of ratings as an x-axis instead of the total number of interactions which was used in Figures 1. The reason for this is that a fair comparison between the two configurations can be made only when the agents in both configurations have the accumulated same amounts of ratings. In configuration 3 the agents rate only 50% of the papers, so it will take them twice as long in average to accumulate comparable number of ratings and trust. The results of this experiment show that when the total rating numbers in the two systems are the same, their performance is almost the same. However, as mentioned above, it takes the agents in configuration 3 longer to build communities.

5. Discussion and future work

The proposed mechanism bears some similarity to a search mechanism proposed by Yu and Singh [17], which uses referral chains to spread queries and locate services. Our mechanism also depends on neighbours to forward queries and find services; referrals and neighbours are all selected based on recommenders' / agents' trust in them. This is an individualistic approach of learning about how to locate good service providers in both mechanisms. However, in addition to this individualistic approach our mechanism provides a collective approach, by forming communities, as a quick way to help agents locate good service providers. A community is analogous to a club in the human society, which attracts people with the same interests and tastes to get together, share their knowledge, learn and benefit from each other. This is particularly useful when an agent is new in an area, since the collective information from the community provides an overview of the available resources, services and agents in the area. In this way a newcomer can learn more quickly than from individual experience. For expert agents, finding information from communities provides a way to follow broader trends in the latest developments, discover new services, ideas, or new potential directions for research that are not covered by the information provided by their most trusted friends. Generally, it is an open question when an expert agent should look for information from its community and when it should get information by asking its most trusted agents (friends).

Our approach to forming communities also bears some resemblance with collaborative filtering, a common technique used in recommender systems [9, 10] where a centralized database is used to store the evaluations of items from users [1]. The recommendation for an item is based on other users' evaluations of the item and the similarity between these users' previous ratings with those of the inquiring user. We use trust to measure another agent's ability of providing good papers, which implicitly includes a measure of the similarity between agents. However, in our approach each agent computes its trust using a simple algorithm and keeps its representation of trust local. There is no centralized database. The community ensures integration of the individually accumulated knowledge, which is a form of centralization, but it emerges as a result of self-organization. Each agent is autonomous and does not depend on the existence of the community. The collaborative filtering approach enforces a community-like effect by collecting all ratings ever given by users in a centralized database and using a complex centralized algorithm for the computation of similarity when a recommendation is needed for a particular user. Therefore, scalability is the main concern.

The question of ensuring incentives to create and join a community is important. It has been pointed out that P2P systems are plagued by "free riders", i.e. people who do not contribute, but only consume resources [13]. In our mechanism, creators are responsible for creating and maintaining communities. Therefore, ensuring that there is an incentive for users to become creators is critical since they have to contribute more resources than other agents. Our file sharing system is built to facilitate people to find valuable papers for their research. For a user, being a creator of a community allows him/her to benefit from the participation of other users who bring their knowledge (ratings or comments about papers) and resources (papers) by saving the creator time and effort to discover information, which otherwise is distributed in the whole system. Another "privilege" of a creator in our mechanism is that it can not be expulsed from a community, even if the trust in it by other peers drops under the normal threshold for trust in community members. This mimics somewhat a phenomenon observed in real communities, where those who start a community (e.g. a conference or journal) may take an administrative role rather than the role of active contributors of services (papers) and still enjoy a respected position in the community.

A third incentive for agents to join communities is that they can become more visible and known than if they are on their own. Although agents who are not creators, even not members of communities, can also benefit from communities, a possible incentive for users to join such trust-based specialized communities is the feeling of excellence since only good peers can join in communities. Authors have incentive to join communities since they can put their publications into the paper list of communities and thus make their papers more visible, since these papers will be recommended and accessed more often.

The way trust is calculated in our mechanism makes it easier to build up trust than to discount the trust in an agent. Negative experiences do not decrease the trust significantly. There have been arguments [15], that a trust mechanism should build up trust slowly, but decrement trust rapidly in order to protect from dishonest agents. However, we believe that in our domain protection from dishonest and malicious peers is not of big concern. Our system is aimed to serve academic communities who share research papers. It is possible that some users provide fake ratings, for instance, a researcher who wants to increase the quoting of certain papers, and decrease the quoting of others. This may bring some glory in the short term, but ultimately the users who are interested in obtaining good papers will leave the community since their trust in the community will drop below their thresholds. If the members however, are willing to participate in a self-admiring community, as sometimes happens in reality, the community will persist and will serve the purpose well.

We found that agents tend to have more interactions with members from their own and from other communities, even though they were set up to search outside of the community, using their random Gnutella neighbourhood in 50% of the cases. This is because when agents download papers from a community, they also increase their trust in the community members. When they search papers in the general agent population and they get results from a member of their community, they are more inclined to choose to download papers from community members since they are trusted more. So even if the community disappears suddenly (e.g. when the creators leave it), the performance of the system will not decrease significantly, since the agents already know which agents are trustworthy. Agents can also reconstruct the community easily, if there is a creator available.

In our mechanism, an agent can not proactively join a community. It has to be invited by the current community members. Although this can prevent some untrustworthy agents from joining the community, it can be a disadvantage since it limits the scope of potential members to those with whom current members have interacted. This "encapsulation" problem that happens also in human communities and therefore "weak ties" are very important to bring information across communities. In our future work, we will try to augment our mechanism with some analogue of "weak ties" and to provide a mechanism for agents to proactively join communities. We will also explore further the statistical significance of the results obtained in the reported experiment in more realistic (e.g. Zipf) distributions of agents, creators, paper ratings etc. and will try to find analytically and/or experimentally good values for the various thresholds used in the trust measures. We will also implement the mechanism in the Comtella system and evaluate it with real users.

6. Conclusions

We propose a mechanism for forming interest-based communities in a P2P system for sharing academic papers using trust and reputation. The communities can provide peers with information about the resources shared in the community and collectively computed rating of their quality as well as about the peers in the community and their reputation. Our simulation shows that such communities can benefit peers.

Acknowledgement. This work has been supported by the Canadian Natural Sciences and Engineering Research Council (NSERC) under the LORNET Project and the Individual Discovery Grant of the second author.

References

[1] Breese, J. S., Heckerman, D. and Kadie, C. "Empirical Analysis of Predictive Algorithms for Collaborative Filtering". *MSR Technical Report-98-12*. May 1998.

[2] Brooks H, C. and Durfee. H. E. "Congregating and Market Formation". In *Proceedings of thefirst international joint conference on Autonomous Agents and MultiAgent Systems (AAMAS'02)*, ppp96-103, Bologna, Italy, 2002.

[3] Comtella. Available to download from the MADMUC lab. University of Saskatchewan (accessed on January 21, 2004). http://bistrica.usask.ca/madmuc/peer-motivation.htm

[4] Gmytrasiewicz, P.J. and E. H. Durfee. "Toward a theory of honesty and trust among communicating autonomous agent." Group *Decision and Negotiation* **2**: 237-258, 1993.

[5] Good N., Schafer J., Konstan J., Borchers A., Sarwar B., Herlocker J., and Riedl J. "Combining Collaborative Filtering with Personal Agents for Better Recommendations" *Proceedings of the 1999 National Conference of the American Association of Artificial Intelligence*, pp 439-436.

[6] Granovetter, M. "The Strength of Weak Ties", *American Journal of Sociology* **78:** 1360-1380 1973.

[7] Li C., Sycara K. "Algorithm for Combinatorial Coalition Formation and Payoff Division in an Electronic Marketplace". *Proc. of AAMAS'2002*, Bologna, pp 120 -127.

[8] Marsella, C. S.; Adibi, J.; Al-Onaizan, Y.; Kaminka, G. A.; Muslea, I.; Tallis, M.; and Tambe, M. 2001. "On Being a teammate: Experiences acquired in the design of RoboCup teams". In *Journal of Autonomous Agents and Multi-Agent Systems*. Volume 4(1-2). "Best of Agents'99" special issue.

[9] McNee S., Albert I., Cosley D., Gopalkrishnan P., Lam S.K., Rashid A.M., Konstan J., Riedl J. "On the Recommending of Citations for Research Papers". In *Proceedings of ACM 2002 Conference on Computer Supported Cooperative Work (CSCW2002)*, New Orleans, LA, 2002. pp. 116-125.

[10] MovieLens. Available on line at http://movielens.umn.edu

[11] Nair, R., Tambe, M., and Marsella, S. "Role allocation and reallocation in multiagent teams: Towards a practical analysis" *Proc. AAMAS'2003*, pp. 552—559, Melbourne, 2003.

[12] Shehory O., Kraus S. Methods for task allocation via agent coalition formation**,** *Artificial Intelligence Journal*, **101** (1-2): 165-200. 1998.

[13] Stefan Saroiu, P. Krishna Gummadi, Steven D. Gribble. "A Measurement Study of Peer-to-Peer File Sharing Systems". *Proceedings of Multimedia Computing and Networking (MMCN) 2002*, San Jose, CA, USA, January 2002.

[14] Tran T. "Reputation-Oriented Reinforcement Learning Strategies for Economically-Motivated Agents in Electronic Market Environments". *PhD Thesis*, University of Waterloo. 2003

[15] Wang Y., Vassileva J. "Bayesian Network-Based Trust Model**"**, *Proc. of IEEE/WIC International Conference on Web Intelligence* (WI 2003), pp. 372- 378, 2003, Halifax, Canada.

[16] Yang B., Garcia-Molina H., "Designing a super-peer network" In *Proceedings of the 19th International Conference on Data Engineering (ICDE)*, Bangalore, India, March 2003

[17] Yu B., Venkatraman M. and Singh P. M, "An Adaptive Social Network for Information Access: Theoretical and Experimental Results", *Journal of the Applied Artificial Intelligenc*, **17** (1): 21-38, 2003.