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

Yao Wang

Department of Computer Science
University of Saskatchewan
Saskatoon, SK, Canada

yaw181@ mail.usask.ca

ABSTRACT

Decentralized P2P networks can benefit from forming interest-based communities which 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 agents.

1. Introduction

Peer-to-peer networks are networks composed of heterogenous and autonomous peers that cooperate with each other in a decentralized manner. All peers are both users and providers of resources and can access each other directly without intermediary agents. Compared with a centralized system, the peer-to-peer (P2P) network provides an easy way to aggregate large amounts of resource residing on the edge of Internet or in ad-hoc networks with a low cost of system maintenance. There are two kinds of architectures in P2P networks, hybrid P2P networks and pure P2P networks. In a hybrid P2P network, there is a server that each peer has to depend on in order to interact with other peers. In Napster, the server provides a global music index and peers have to visit the server first to know the peers that share a given file before downloading the file from some of them. In the pure P2P network, like Gnutella or Freenet, there is no server. Each peer has to find other peers' shared resources by itself. The two kinds of P2P networks have some advantages and disadvantages. A hybrid P2P network is more efficient than a pure P2P network since it saves the effort of peers finding all the information individually. However, it is vulnerable because when the server fails, the whole system will crash. On the other hand, maintaining a server is costly. In a pure P2P system those costs are shared by the peers, who have to find out all the information by themselves in a costly way. For example, Gnutella peers use a flooding algorithm to find the resources, which consumes a lot of communication and computation. More important is that peers have to spend the same effort individually to find out the same information, such as what peers have what resources and whether a peer is reputable. This causes inefficiency and waste of resources, such as computer power and network bandwidth, but has the advantage that the system is scalable and robust. The failure of any peer will not influence the whole system. The disadvantages of the two kinds of P2P networks are undesirable in the situation where efficiency and robustness are both important. Forming communities in a pure P2P network could provide a way to solve the problem. In the networks, peers can get together to build communities. The communities can provide peers with collective and authoritative information, which makes them act like servers as a whole. But peers are autonomous. Even without communities, they can still perform as they do in a pure P2P network.

We propose a mechanism for forming communities in a P2P system for sharing academic papers. The rest of this paper is organized as follows: section 2 discusses related work in the area of agent group-formation. Section 3 describes the P2P application in which the community mechanism applies. Section 4 presents our approach to forming communities. The experiment design and results are described in section 5. The last section discusses some related issues about our mechanism and directions for future work.

2. Communities

The word "communities" has been widely used in the literature recently. While some authors mean virtual communities of real human users, the communities referred to in the context of multiagent systems are groups of agents. Different ways of forming communities have been proposed. Generally, we can classify them into three levels. At the low level, a community is simply referred to a group of agents that cooperate with each other in the same environments, such as "eCommerce communities" [29], "Electronic Communities" [31], "Peer to Peer Communities" [26]. In such contexts, a whole system is a big community composed of all of the agents. Communities at the median level are referred to a group of agents that tend to communicate or interact with each other more often. The communities are automatically and implicitly formed by agents to facilitate their cooperation. Different communities can be identified according to the criteria, such as the link topology [5, 6, 9], the interests of agents [13], and the neighbourhood of agents [30]. In such communities, agents are just physically grouped together. The agents in the same communities do not work as a whole to achieve some common goals that will benefit to all the members. At the high level, the community is defined as an organization that facilitates a group of agents that share common interests and preferences to get together, share their knowledge, learn and benefit from one another. The communities are formed explicitly and purposely. All agents in the same community will work as a whole, both contributing to and benefiting from the community, analogous to human community, such as the online communities, *friendster* [33] and *tribe* [35]. At this level, there are several similar concepts in multiagent systems, such as coalitions, teams, congregations. Although they are all targeted for the cooperation among agents, they are different from each other.

- *Coalitions* are mostly used in e-commerce or utility-related fields and composed of self-interested agents. The research on coalitions mainly focuses on how to form coalitions that maximize the group utility and distribute the utility among the members. Game theory methods are used in analyzing and solving the problem. The motivation for agents to join coalitions is that they can get more benefit in coalitions, although they can also act alone. The typical scenario for coalitions is that a group of buyers forms a coalition to get some price discount or benefit [1, 14, 21, 22, 23].
- *Teams,* like robot soccer teams, are composed of agents cooperating to solve a problem that cannot be solved by any individual member or to solve a problem more efficiently. The main focus of teamwork is the assignment of tasks and the coordination among the agents [12, 15, 18, 20].
- *Congregations* are long-term groupings of self-interested agents, formed by self-organization [3]. Congregations divide a large group of agents into subgroups composed of agents of the same type or with the same interests who would prefer to interact with each other. Although agents can move from one congregation to another, they can only interact with agents in the same congregation at a given time. The benefit of forming congregations is that agents can more easily find each other and can have more successful interactions with the agents in the same congregation. Agents in congregations are more loosely coupled than the agents in coalitions or teams. They do not necessarily have common goals.

We define an agent *community* as an organization that facilitates a group of agents that share common interests and preferences to get together, share their knowledge, learn and benefit from one another. A community of agents has similarities with all types of agent groups described above. Agents in a community all work together to achieve some common goals, which can facilitate achieving their individual goals. They can also act alone. But the goals of a community are long-term and 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. Also, while in coalitions and congregations agents interact only within their group, or with other agents as a group, 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. A P2P System for Sharing Papers

People typically find research papers by searching for key words in Google or Citeseer. After reading a paper, they often evaluate the paper and have questions and opinions about it. If people are interested in the same area, most often they will read the same papers. Sometimes they may want to talk to each other about the papers and share their opinions, which could help them understand the paper better. But there is no way for them to know each other and share opinions. A P2P file sharing system may be a good way to solve the problem. In the system, people share not only the papers that they are interested in, but also their evaluations and comments about the papers. So when people are looking for papers in the system, they can also read other people's ratings and comments about the papers, which helps them to decide whether a paper is worth to read. They can also compare their opinions later and learn from each other. This could benefit the authors of the papers. They can know how other people think about their work. They may get some valuable suggestions or inspiration from the comments of other people. A P2P file sharing system for research papers called Comtella [4] has been developed for this purposes at the MADMUC Lab in the University of Saskatchewan, which 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.

Although people can greatly benefit from these advantages, these benefits could be traded off by the problem of information overload. For example, when people search for papers, they can get a long list of papers. Although people can read the ratings of the papers, people may rate a paper differently since they have different interests and knowledge. A person not interested in agents may give a bad rating to a high-quality paper on agents. A person with little knowledge of artificial intelligence may rate a low-quality paper on artificial intelligence highly. So it is still hard for people to decide which paper is good. Since agents are often used to assist people in P2P networks, forming communities of agents that can recommend good papers may be a way to solve the problem. In the communities, agents share their information and help each other to find good papers. Here we propose a self-organizing mechanism for agents to form communities.

4. 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 an 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.

4.1 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? In order to create a community, users need to contribute more resources, such as computer processing time, disk space, and bandwidth, so that their agents can construct a community, compute and store the collective information related to the community.

Only the agents of users who want to contribute resources can form communities. We call these agents *creators*.

2. Who will be eligible to be a member of the community? Some users share good papers and rate papers competently and fairly. Some may be free riders, who do not share papers or never rate papers. For a self-sustaining 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 is the responsibility 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 the 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 of its 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 can invite 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 can also invite its own set of trustworthy agents in the category to join the community. In this way, the agents in the community can help each other to find other potential trustworthy agents and the community can grow quickly. An agent can join a community only when it is invited by one of the community members. So each agent in a community, except the creator, is trusted at least by one community member.

An agent can leave a community freely, if the community becomes no longer trustworthy. Each community can only have limited community members. The maximum number of members in the community is called *community capacity* and is 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 disk space and CPU can be dedicated to serving the community. When a new creator joins a community, the community capability will increase as a function of the number of creators and the resources they contribute. The resources contributed by the creator(s) are used to hold a directory of all shared resources in the community, the ratings of these resources and to compute the trust in community members and other peers who interact with the community.

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 [11]. When multiple creators work in the same community, they share their resources and organize themselves to provide a centralized view to all community information. The next sections will discuss the metrics for individual trust in another agent, collective trust in a community, the mechanisms for agents updating their neighbours, and the update mechanism for communities.

4.2 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, [25] which emphasizes that the performance of the trusted person/agent has to meet or exceed the expectations of the trusting agent, and [27] which emphasizes similarity in judgment criteria between the trusting and the trusted person/agent. Other definitions emphasize different aspects of trust, for example [10] 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 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 usings 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 original provider and provider of the paper, respectively, when its user rates the downloaded paper. We assume that users rate the papers using a 5 scale rating scheme, from “1” (worst) to “5” (best). Trust is updated using the following reinforcement learning formula.

$$trust^n = \alpha * trust^o + (1-\alpha) * e_\alpha \qquad (1)$$

$trust^n$ denotes the trust value after the update; $trust^o$ denotes the trust value before the update. α is the learning rate – a real number in the interval (0,1). e_α is a value of 0 or 1. Giving a threshold *t_rating* and the user's rating, *rating*, if $rating \geq t\_rating$, which means the user is satisfied with the paper, e_α equals 1 and the user's trust in the provider increases; if $rating < t\_rating$, e_α equals 0, which means the user is unsatisfied with the paper and his trust decreases accordingly. Given a threshold *tt*, a provider is trustworthy for a user if $trust \geq tt$. Otherwise, the provider is not trustworthy.

In the system, the original providers are important since they introduce new papers. The new papers will be propagated to other users later. 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.

4.3 Trust in a Community

An agent builds its trust in other agents based on its interactions with them. Analogically, an agent can build its trust in a community using its experiences with the community. When the agent gets a good paper from a community, it can increase its trust in this community; otherwise, it can decrease its trust. It will be possible to build trust in a community in this way if the community stays stable for a long time. But in our mechanism, communities can be constructed or destroyed quite often. Once a community disassembles, the agent's previous experiences with the community will become meaningless. If a new community is formed, the agent has to build its trust in the community from scratch, which is time and effort consuming. So another alternative is using the average trust of the agent in each community member to measure the agent's trust in the whole community. So no matter how the communities change, the agent can always quickly know whether the communities are trustworthy.

$$trust\_community = \frac{\sum_{q=1}^{r} trust_q}{r} \tag{2}$$

trust-community denotes the agent's trust in the community. *r* is the number of the community members. $trust_q$ is the agent's trust in the *qth* community member. Given a threshold *tc*, if $trust\_community > tc$, the community is trustworthy for the agent. Otherwise, it is not trustworthy.

4.4 Community Update

Each community is an entity composed of members and resources as shown in Figure 1. The purpose of the communities in the P2P system is to bring good agents together, collect information from them, and guide agents to find good papers. There are three lists in each community, the creator list, the agent list, and the paper list. The creator list stores the information about the creators, including the creator IDs, IP addresses, and the information about their shared resources. The agent list includes the information about agents, such as IDs, IP addresses, reputations, membership. The paper list is used to store the information about the papers shared by the members, such as paper titles, ratings and links to the paper providers. All the lists are stored by creators. Other agents can get the information from a community by connecting to one of its creators.

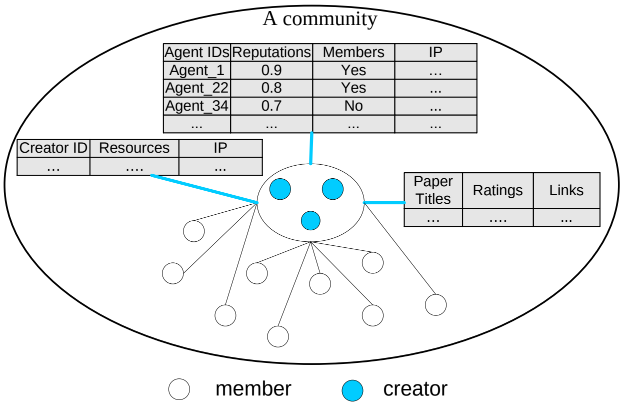

Figure 1. The structure of a community

In our mechanism, any community member can introduce new agents into the community without the consensus of other members. This is a quick and easy way to introduce a new member into the community. But it is also easy for community members to bring bad agents into the community by mistake. So a community needs to find these bad agents and expulse them. Another reason for updating the members of a community periodically is to adapt to the changes of the agents. A community member may have behaved very well and provided a lot of good papers before joining the community, but after it joins the community, it may start to behave badly by providing garbage papers or even spreading viruses.

An agent's *reputation* within a community can be used to judge whether the agent is good or bad. It is a collective measure of how much the agent is trusted by the members of the community. When updating the reputations of its members, the community will ask its members to evaluate the agents that they have ever interacted with. Then it will compute the reputations of these agents as follows (3):

$$R_i = \frac{\sum_{j=1}^{l} trust_{ji}}{l} \tag{3}$$

R_i denotes the reputation of the *agent i,* which is the average trust of the community members in the *agent i*. l is the number of the community members. $trust_{ji}$ is the trust of the *jth* evaluating member in the *agent i*.

The community sorts all the agents by their reputations and asks the top agents with the best reputations to join it if they are not in the community and their reputations have to be over a given threshold. This is how a community purges itself from members who have become less trusted. The creators can not be expulsed from a community since they contribute resources that the community needs. A creator can leave a community when the community is no more trustworthy. 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 dropped from the community.

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

$$rating\_community = \frac{\sum_{z=1}^{g} rating_z}{g+1} \quad (4)$$

rating_community denotes the rating of a paper from the community. $rating_z$ is the *zth* rating of the paper from the community members. *g* is the number of the ratings of the paper.

4.5 Neighbors Update

In our system, neighbors are referred to the agents to whom a given agent sends its queries directly. The number of an agent's neighbors is limited. The neighbors are important for an agent since they determine the other agents that the agent's queries can reach. In our mechanism, after an agent finishes an interaction, it will update its neighbors by choosing its most trusted agents. By this way, an agent can find more good papers since the agent sends queries to its trusted agents, who will forward the queries to their trusted agents (their neighbors).

4.6 Application for joining a community

A snowball effect can appear in human societies, e.g. paper citations, and in multiagent systems when agents recommend trustworthy agents to each other. A reputable agent becomes more reputable because of the recommendations among agents and therefore attracts more agents to interact with it, which gives more recommendations for it etc. But our previous work [28] shows the side effect of this phenomenon that agents tend to know the same set of trustworthy agents and interact with them repeatedly. The same phenomenon happens in this system, which causes some good agents who happen to be in communities earlier overloaded. A lot of agents download files from them. On the other side, some agents are ignored and no agents download files from them even if they provide files as good as those overloaded agents do. So these ignored agents have no chance to become trusted and no chance to join a community. To avoid this, we have to modify the mechanism to give every good agent a chance to join a community. When an agent finds a trustworthy community and wants to join the community, it can send its application to one of the creators in the community and recommend several papers to the community. These papers should be unique, which means they are only shared by the applying agent, not by any members of the community. When the community creator receives the application of the agent, it will put the agent's recommended papers into a separate list. The members of the community can randomly select papers from the list, judge them, and build trust in the paper provider. If the papers are good, they will be recommended to more agents. The paper provider can also be known by more agents. Therefore it can get a chance to be trusted by a community member and to be invited to join the community.

5. Simulation and Experiments

To evaluate our approach, we developed a simulation of the P2P file sharing system using JADE 2.5. For different categories, an agent builds different trusts in other agents and has different neighbours to send queries. An agent's trust in another agent in one category will not influence its trust in the same agent in another category. The formation of communities in one category will also not influence the communities in another category. So, for simplicity, we just used one category in the simulation.

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 1 to 5. 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 neighboring peers in the way queries are sent in Gnutella. In our experiments, agents will search papers in their own communities with a 50% chance if such communities exist. When agents search papers outside their own communities, they will search papers in other communities with a 50% chance and from the general population with a 50% chance. If the agent is searching for a paper in a community, it will always choose the most highly recommended paper by the community. Since users can share papers with or without ratings, when an agent selects papers from the general population, it will first choose the paper with the best rating *mv* from the rated papers according to the formula (5).

$$mv = trust * rating \quad (5)$$

trust is the agent's trust in the provider's capability of providing good papers, obtained from the formula (1). *rating* is the rating of the paper from the provider. Given a threshold *tp*, if $mv < tp$, 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*.

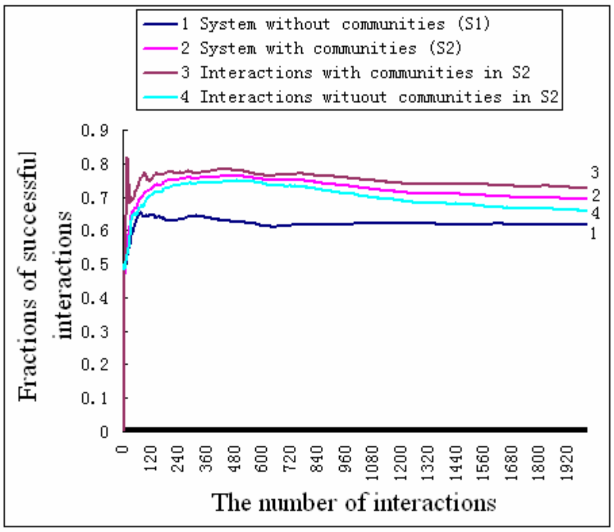

Figure 2. The average of the performances in systems with and without communities

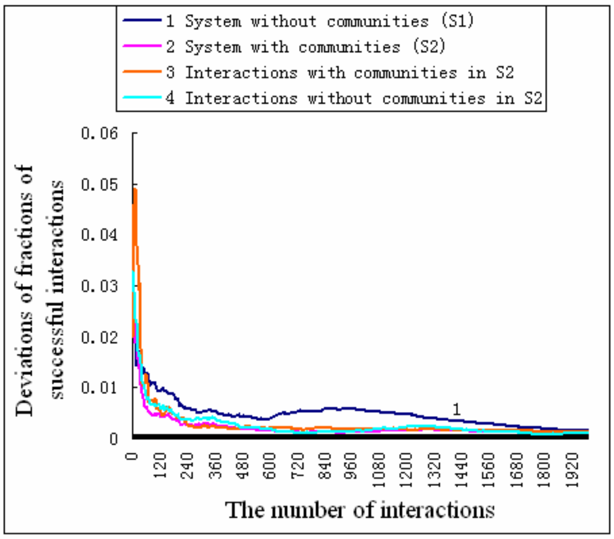

Figure 3. The standard deviation of the performances in systems with and without communities

The goal of the first experiment is to see if forming communities helps agents find good papers. We compare the two systems with communities (S1) and without communities (S2). In the two systems, all the papers are not rated at the beginning. The threshold *tt* for an agent to be trustworthy is 0.53. If an agent downloads a paper and gives a rating over 3, we call it a successful interaction. If the paper is selected from a community, we call it an interaction with communities. Otherwise, we call it an interaction outside communities. We compare the overall performances of the two systems in term of fractions of successful interactions. In the system with communities, we also measure the performances of the interactions with communities and outside communities. Figure 2 and Figure 3 shows the average and deviation of the fractions of successful interactions in the two systems, respectively. We can see that the overall performance of the system with communities is better than the system without communities. In the system with communities, the performance of interactions outside communities is also better than the performance of the system without communities, which implies the existence of communities influences agents' selection of papers even when agents just use their own knowledge, not the collective knowledge from the communities, to choose papers. The reason is that the agents tend to select papers from their trusted agents. The communities help agents locate their trustworthy partners more efficiently and successfully when agents interact with them, which will benefit the agents in their further search for good papers even without the assistance of communities.

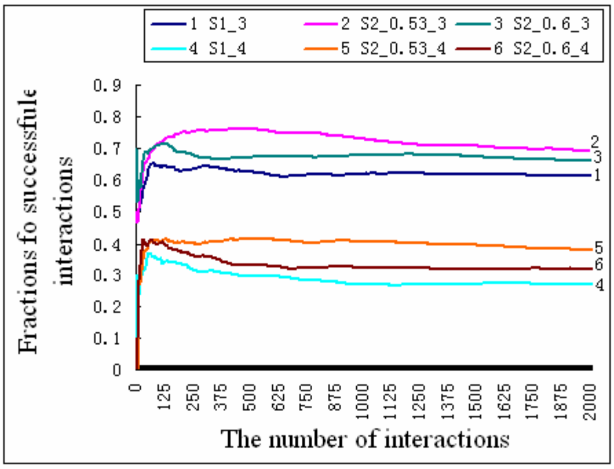

Figure 4. The influence of the criteria of evaluating interactions and agents

The goal of the second experiment is to see how the different criteria of evaluating the interactions and the trust in agents influence the performances of the two previous systems S1 (without communities) and S2 (with communities). In this experiment, we use 3 and 4 as the threshold, *t_rating*, for evaluating a successful interaction, respectively. The threshold for judging a trustworthy agent, *tt*, is also raised from 0.53 to 0.6. Figure 4 shows that no matter whether *t_rating* is 3 or 4, the performance of S2 is better when *tt* is 0.53 than when *tt* is 0.6, which means that a high threshold for judging trustworthy agents is not good for the system. In our mechanism, communities are acting like information centers to collect and spread information among agents about good papers and good agents. A higher threshold *tt* will make agents hard to regard other agents as trustworthy agents and establish communities with them; therefore the advantage of communities cannot be exerted fully. An appropriate threshold for judging trustworthy agents is important for achieving good performance.

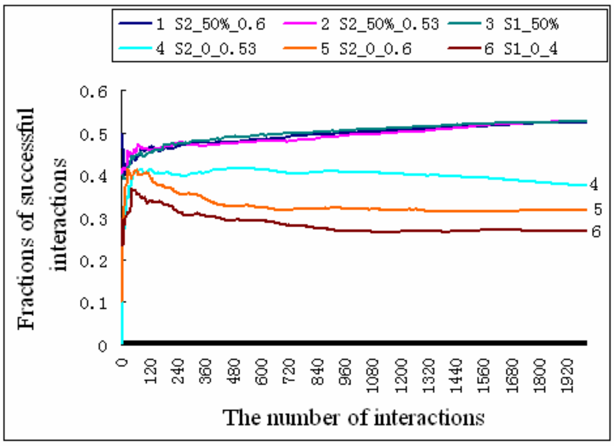

Figure 5. The influence of the initial ratings

In the two previous experiments, all the shared papers are not rated initially. So the agents just select papers blindly at the beginning. But in a more realistic situation, some people may share their ratings when they share papers. The system will have some rated papers at the beginning. So in the third experiment, all the agents rate 50% of their shared papers at the beginning. *t_rating* equals 4. *tt* equals 0.53 and 0.6, respectively. The goal of this experiment is to see how the initials ratings influence the system. Figure 5 shows that the systems where 50% of the shared papers are rated initially perform much better than the systems without any initially rated papers. The reason is obvious: the providers' ratings can guide other agents to find good papers efficiently. Figure 5 also shows that the system forming communities cannot perform better than the system without communities when 50% of the shared papers are rated at the beginning. So forming communities is much useful in the systems that have few paper ratings.

6. Discussion and Future work

In our experiments, 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 communities, using their Gnutella neighborhood 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 the community members, they are more inclined to choose to download papers from the 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.

Our method of forming communities is similar to the method used to form on-line communities, such as *friendster* [33], *tribe* [35] and *orkut* [34]. In these communities, a new member can join a community only when s/he is invited by some member inside the community. For example, in the community of *friendster*, the users inside a community can invite their friends to join the communities so that everyone in the community can know their, and their friends' friends, meet or date with them. These communities are supposed to be composed of good members (since they are trusted friends). But an online community is always vulnerable to bad members. Once a bad member is introduced into a community, it is very hard for the community to get rid of him/her. A bad member can also bring other bad members into the community, and form a clique. Finally the whole community will be ruined [8]. However, in our mechanism, a member can be expulsed from its community if it has a bad reputation in the community.

In our mechanism, when an agent sends queries to its neighbours to search papers, its neighbours will forward the queries to their neighbours. Since an agent's neighbourhood is composed of its most trusted agents, we can view the neighbours of the agent as *recommenders*, who recommend their neighbours to the agent. These recommender agents are called "*referrals*" in Yu and Singh's mechanism [32]. So the agent implicitly asks the *recommenders* / *referrals* for answers when its queries are spread by its and its neighbours' neighbours. But in Yu and Singh's mechanism agents ask the referrals for answers explicitly. In contrast, our mechanism is lightweight in communication between agents – each recommender directly forwards the queries to its own *referrals*, while, in Yu and Singh's work, each *recommender* has to send back the list of its *referrals* to the inquiring agent, who will send the queries to the *referrals* later. In Yu and Singh's work, the agents measure other agents' abilities of providing services and their sociability, i.e. their abilities to find the agents that can provide good services. This is an individualistic approach of learning about how to locate good service providers. Forming communities is an alternative mechanism that we propose as a quick way to help agents locate good service providers. A community is analogous to a club in our human society, which attracts people with the same interests and tastes to get together, share their knowledge, find each other, learn and benefit from each other. Especially when an agent is new in an area, the collective information from the corresponding community provides an overview of the available services and agents in the area. In this way a newcomer can learn more quickly than from extensive individual experience. For expert agents who have a lot of knowledge and know many other agents, finding information from communities provides another way to follow the latest developments, discover new services, or ideas, or new potential directions for research that are not covered by the information provided by their trusted friends.

Generally, when expert agents should look for information from communities and when they should get information from their friends is an open question. In Yu and Singh's work, it is unclear how an agent selects the best service for evaluation from all the suggested services. Usually it is impractical that an agent will evaluate all the suggested services for a particular query since it is not only time-costly, but also an agent has no motivation to do that once it gets the best service. In our mechanism, we suggest a simple algorithm to find the best service using inquiring agents' trust in the service providers and evaluations of the services from the service providers if their evaluations exist. The metric may be different in different contexts.

Our approach for forming communities bears some resemblance with collaborative filtering, a common technique used in recommender systems. The memory-based algorithms are the most popular algorithms used in most recommender systems. The algorithms include content-based filtering [7, 16], user-based filtering [17, 20] or hybrid filtering [11]. Most of the recommender systems are centralized. The GroupLens research group suggests a mechanism for recommending research papers [16]. It uses citations in research papers to measure the similarity of papers and make recommendations. For each recommendation, a lot of computations have to be performed in real time using a centralized database. Any previous computation result is useless for a new computation. For such centralized recommender systems, scalability is the main concern. Although a decentralized recommender system is suggested by Olsson [19], its algorithm for recommending items is similar to that used in the recommender system of research paper except that agents only models part of other agents, not all the agents in the system. All memory-based recommender systems follow the typical mechanism described in [2], where a database is used to store the evaluations of the items from the users. The recommendation for an item is computed according to other users' evaluations of the item and the similarities between the inquiring user and the other users who have evaluated the item already. The resulting recommendations are personalized. In contrast, in our mechanism, the recommendations from communities are not personalized, but rather reflect the opinions of the majority of the community about papers or agents. Therefore, they can guide other agents, especially newcomers, to find good papers and agents. Agents can also get personalized recommendations by sending queries to their most trusted friends or friends' friends and asking them for recommendations. Although using personalized recommendations can help agents find results more accurately matching their preferences, community recommendations can help agents find answers in a broader scope since communities consist of more heterogeneous agents and therefore they may have knowledge of papers or agents not known by the friends of the inquiring agent. Our method is also different from that used in collaborative filtering. We use trust to measure another agent's ability of providing good papers, which in some sense includes a measure of the similarity between agents. However, each agent computes its trust using the learning formula and keeps its representation of trust local, without using a centralized database. The community ensures synergy of this individually accumulated knowledge. The collaborative filtering approach enforces a community-like effect by collecting all ratings ever given by users in a centralized database and using a very complex centralized algorithm for the computation of similarity when a recommendation is needed for a particular user. In our approach the computation of trust much simpler. Also each agent is autonomous and does not depend on the existence of the community. The collaborative filtering approach is impossible without the centralized server and database.

In our mechanism, creators are responsible for creating and maintaining communities. It has been pointed out that P2P systems are plagued by "free riders", i.e. people who do not contribute, but only consume resources [24]. 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 proper and valuable papers during their research. For a user, being a creator to create a community in his research area is of great benefit to his research. Once a community is created, other agents with similar interests and tastes will be attracted. So the creator can benefit from the participation of other agents who bring more knowledge (ratings or comments about papers) and resources (papers). The information provided by communities is so structured and organized according to interests, paper ratings and agents' reputations that it can save the creator time and effort to discover such 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 mechanism observed in real communities, where those who start a community may take a more administrative role than the role of active contributors of services (papers) and still enjoy a respected position in the community. The 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.

In our previous experiments, we model a system where 50% agents are creators. When we reduced the number of creators to 20%, the system performance does not show obvious improvement compared with the system without communities. Our future work will focus on discovering what is the critical mass of creators and "early adopters" needed to form successful communities.

References

[1] Breban S., Vassileva J., "A Coalition Formation Mechanism Based on Inter-Agent Trust Relationships". To appear in L. Johnson & C. Castelfranchi (eds.) *Proceedings of the First Conference on Autonomous Agents and Multi-Agent Systems*, Bologna, Part 1, ACM Press, 306-308, 2002.

[2] Breese, J. S., Heckerman, D. and Kadie, C. "Empirical Analysis of Predictive Algorithms for Collaborative Filtering". MSRTR-98-12. May 1998.

[3] Brooks H, C. and Durfee. H. E. "Congregating and Market Formation". In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems (AAMAS)*, page 96-103, Bologna, Italy, 2002.

[4] Comtella. Available to download from the MADMUC lab. University of Saskatchewan (accessed on January 21, 2004). http://bistrics.usask.ca/madmuc/peermotivation

[5] Flake W. G., Lawrence S., and Giles L.C., "Efficient Identification of Web Communities," *Proc. 6th Int' Conf. Knowledge Discovery and Data Mining*, ACM Press, New York, 2000.

[6] Flake,W. G., Lawrence, S., Giles, C. L. & Coetzee, F. "Self-organization and Identification of Communities". *IEEE Computer* **35,** 66 - 71 (2002).

[7] Foner, L.N. 1997. Yenta: "A Multi-Agent, Referral-Based Matchmaking System". *In Proceedings of The First International Conference on Autonomous Agents*, 301-307. ACM Press.

[8] "Friendster Quickly Gathering Foes". http://www.wired.com/news/culture/0,1284,61150,00.html

[9] Gibson D., Kleinberg J., and Raghavan P. "Inferring web communities from link topology". I*n Proc. 9th ACM conference on Hypertext and Hypermedia*, 1998.

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

[11] 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.

[12] Kaminka, G. A.; Pynadath, D. V.; and Tambe, M.. "Monitoring Deployed Agent Teams", In *Proceedings of the Fifth International Conference on Autonomous Agents (Agents-2001), Montreal, Canada,* 2001.

[13] Khambatti M., Dasgupta P. and Ryu D.K. "A Role-Based Trust Model for P2P Communities and Dynamic Coalitions". To appear *IEEE Int'l Information Assurance Workshop (IWIA)*, Charlotte NC, April 2004.

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

[15] 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.

[16] 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.

[17] MovieLens: movielens.umn.edu

[18] Nair, R., Tambe, M., and Marsella, S. "Role allocation and reallocation in multiagent teams: Towards a practical analysis" *Proceedings of the second International Joint conference on agents and multiagent systems (AAMAS*), 2003.

[19] Olsson T. "Bootstrapping and Decentralizing Recommender Systems", *Licentiate Thesis* 2003-006, Department of Information Technology, Uppsala University and SICS, 2003

[20] Resnick, P., Iacovou, N., Sushak, M., Bergstrom, P., and Riedl, J. "GroupLens: An open architecture for collaborative filtering of netnews". *Proceedings of the 1994 Computer Supported Collaborative Work Conference*. (1994)

[21] Shehory O., Yen. J, Ioerger T. and Vassileva J. "*Teamwork And Coalition Formation*" AAMAS'02 workshop.

[22] Shehory O., Sycara S., and Jha S. "Multi-agent Coordination through Coalition Formation", Lecture Notes in Artificial Intelligence no. 1365, *Intelligent Agents IV*, A. Rao, M. Singh and M. Wooldridge (Eds.), pp 143-154. Springer, 1997.

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

[24] 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. Received Best Paper Award

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

[26] Vassileva, J. "Motivating Participation in Peer to Peer Communities", *Proceedings of the Workshop on Emergent Societies in the Agent World, ESAW'02*, Madrid, 16-17 September, 2002.

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

[28] Wang. Y, "Bayesian Network-Based Trust and Reputation Model in Peer-to-Peer Networks". *Technical report*.

[29] Xiong L., Liu L. "A reputation-based trust model for peer-to-peer ecommerce communities". *ACM Conference on Electronic Commerce 2003*: 228-229

[30] Yolum P., Singh P. M. "Emergent Personalized Communities in Referral Networks". *IJCAI Workshop on Intelligent Techniques for Web Personalization* (ITWP), 2003.

[31] Yu B. and Singh P. M., "A Social Mechanism of Reputation Management in Electronic Communities", *Proc. of the 4th Int'l Work. on Cooperative Information Agents*, M. Klusch, L. Kerschberg (Eds.), *Lecture Notes in Computer Science*, vol. 1860, Springer, 2000.

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

[33] www. friendster.com

[34] www. orkut.com

[35] www.tribe.net