

# Supporting Peer-to-Peer User Communities

Julita Vassileva

Computer Science Department  
University of Saskatchewan  
57 Campus Drive  
Saskatoon, Saskatchewan, S7N 5A9 Canada  
jiv@cs.usask.ca

**Abstract.** The paper describes a design of a peer-to-peer system which is being developed currently to support file and service (help, advise) sharing in research groups and groups of learners. The design addresses some non-technical problems in the deployment of P2P systems, such as coping with free riders, creating closely-knit groups of users sharing common interests and trust. It describes a user modelling approach for servants and presents several approaches for motivating users to participate and contribute to the community.

## 1. Introduction

In our experience in designing and deploying on a large scale a multi-agent system for peer help called "I-Help" [18], we encountered many issues that are important in the design and practical usage of Peer-to-Peer (P2P) systems. Instead of file sharing, I-Help allows exchange of help and advice services, or "peer-help" among the students. A student needing help related to assignment in a class can request it through her agent who finds other students who are currently on-line and have expertise in the area related to the question. Thus, the help request can be considered as an analogue to a search query in a P2P system and locating an appropriate helper can be considered equivalent to a "hit". In I-Help, agents act on behalf of the users instead of servants, and there is a centralized matchmaker service, which maintains models of the user's competences and matches them to the help-requests. Thus, the I-Help architecture is similar to Napster, which is generally considered a P2P system [12]. The actual service (peer-help) is provided through a chat session, which is recorded. If both users have evaluated it positively and agree to make it public, the recorded session becomes a resource that is available to other students in the system, who come with the same or a similar question. I-Help has been deployed for two years in most undergraduate classes at the University of Saskatchewan and has taught us a lot of lessons [7,18] that are relevant to peer-to-peer systems in general.

Most of these lessons concern non-technical problems in the deployment of P2P systems, such as coping with free riders, creating feelings of trust in the users and motivation to participate and contribute to the community. One major lesson we learned is that the success of such a system requires a "critical mass" of users to participate at each given time. To achieve this it is necessary to know more about the users, i.e. to create models of the users and of their emerging social relationships with other users in the system. We also learned some important lessons about different ways to motivate participation and contribution to the community.

We are currently re-implementing I-Help as a peer-to-peer application, called COMUTELLA (Community Gnutella). The system will enable research or study groups of students to collaborate and share resources, e.g. to exchange both services and files. The goals are:

- to increase the performance, i.e. the speed and quality of search (in terms of precision / recall);
- to increase the satisfaction of the users, their participation and level of contribution.
- to provide a possibility for tightly-bound interest groups of users and to ensure fast routing of queries to the appropriate groups.

Therefore this paper does not report on a piece of finished work; it rather describes the design of a system that is being developed right now, and which draws on our previous experience and some knowledge of P2P computing. The remainder of the paper is organized as follows: section 2 presents a motivation for group formation in P2P systems, section 3 presents the user modeling that needs to be done to allow this, section 4 discusses various methods of motivating users to participate actively, section 5 explores some aspects of the global behavior of the system and section 6 presents a discussion comparing this approach with other work.

## 2. Intelligent Routing and User Interest Groups

Intelligent routing and network organization in Peer-to-Peer networks has been a topic or active research recently [12]. The most prominent approach for document routing, used in FreeNet involves moving data in the network and storing it at certain nodes (servents) to optimize performance. This approach ensures privacy and contributes to a very efficient routing in large, global communities. However, it causes loss of information about the application and the locality [9] and is prone to splitting the network into "islands" [12].

More recently, approaches that try to exploit the social interactions between peers have been proposed. Local search strategies introduced in [1] use well-connected servents and have costs that scale sub-linearly with the size of the network. Ramanathan et al. [13] propose to modify the approach of query broadcast in a random servent neighborhood adopted by Gnutella. They propose selecting the set of neighbors among those that have the highest number of query-related files (hits). Thus a new neighborhood of the servent will be defined for the current search, which contains the most promising at the moment neighbors. This reduces the number of query messages sent in the network by both reducing the number of servents to which a query is broadcasted and the time-to-live (TTL) of each query. This approach has no

“memory” of which nodes were returning many hits about a particular area in a past session. It adapts reactively to the current search, and in order to achieve a better adaptation of the neighborhood, it requires that the users make several consecutive queries in the same general area of search. The approach will not work if the user searches for a given thing just once, or if the user searches for two different things in parallel.

Another approach for semantic routing of queries was adopted in NeuroGrid [8]. It assumes that nodes / servents will store files according to the interests of their users and will persist in the network. The main idea is that each servent stores associations between keywords and other NeuroGrid servents that have returned hits for these keywords in the past. A new query is forwarded to a subset of servents that historically have shown to possess matches to the keywords in this query. This approach makes a shift towards modeling long-term characteristics in the files and nodes to facilitate search. These characteristics are the semantic of the files/resources and associating servents / nodes with particular semantics. One possible criticism is that the network is so dynamic, that there is no guarantee that the nodes with resources matching a particular keyword will be available at the time of the search. However, if they are not available, NeuroGrid can always rely on the default Gnutella neighborhood, i.e. there is no loss, but there can be win, in case these nodes are on-line. NeuroGrid makes a first step towards modeling similarity in interests of users, but this is done implicitly and neutrally, for all search queries passing through each servent. We believe that modeling the interests of users explicitly and maintaining lists of “friends” with similar interests may reap more benefits, since it allows taking into account similarities in patterns of time on-line. We propose that the servent should use a model of its user’s interests and relationships in section 2.

An extension to the Gnutella protocol was proposed in [3] which allows servents to exchange reputation information. The subjective reputation is learned from their previous experience with other servents and reflects the quality of resources and service. There is no semantics associated with the reputation, i.e. “node x is good in area y”. Thus modeling the servent’s reputation does not benefit the search, but assists the user to decide whether to download a file from a given servent, if there are several hits available.

A further step towards facilitating search and selecting quality files/services is to provide mechanisms for servents in P2P systems to self-organize into *groups* based on similarity of interest between their users and on shared positive experiences reflecting quality of resource, speed of download or quality of service.

Many Internet communities, like newsgroups, chat-rooms, and virtual cities constitute examples of self-organizing groups of users with similar interests. Their success could be explained with the high value (or usefulness) of networks that allow group formation. Such networks are known as Group-Forming Networks (GFN). In general, the value of a network is defined [14] as the sum of different access points (users) that can be connected for a transaction for any particular access point (user) when the need arises. There are three categories of values that networks can provide: a linear value, a square value, and an exponential value. The Sarnoff’s Law [15] that states that the power of a broadcasting network is linearly increasing in proportion to the number of its users. Examples of broadcasting networks are TV channels or news sites.

The Metcalfe's Law [11] that states that the value of a peer-to-peer network is proportional to the square of the number of its users. Metcalfe's law is applicable to networks such as telephony systems or P2P networks and has been used by economists as an explanation for the fast growth of the Internet.

Reed [14,15] finds that the networks that allow group affiliation are even more powerful. According to his GFN Law, the value of group-forming networks grows exponentially with the number of users. As a consequence, networks that allow group formation among its components (users or agents) are expected to bring the highest economical benefit.

We envisage a group forming mechanism for Peer-to-Peer networks, based on user's long-term interests and common patterns of behavior. We provides servants with the capability to learn not only about the user's interests, but also to establish relationships with other servants, based on the user's compatibility in interests and satisfaction from previous interactions. It allows for creating long-term relationships between users, takes into account the balance of taking / giving from the community, thus creating a "small-world" – a tightly coupled community based on shared interests, where we believe mechanisms for motivating user's participation can be successfully incorporated.

### **3. User Modeling**

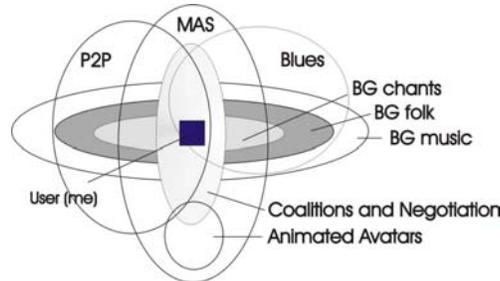
To create user groups based on interest, the servant needs to understand the interests of the user and facilitate finding and maintaining relationships with users with similar interests. In the next sections we will explain how this can be achieved.

#### **3.1. User Model Representation**

The user model contains three different parts:

- a model of the user's interests,
- a model of the user's resources (files or services),
- a model of the user's relationships.

The model of user interests is represented as a list of topics / areas in which the user is interested. An ontology representing topics and sub-areas in a given semantic area allows clustering users into groups sharing similar interests. It is important to note, that these user groups can overlap on various levels, e.g. one user can be a member of a group interested in Bulgarian folk music and in a group interested in blues. The same user can be also a member of a group interested in peer-to-peer computing and in a group interested in multi-agent systems. However, s/he may be a member of a sub-group of the multi-agent systems group interested in agent negotiation and coalition formation and not be a member of a group interested in animated avatar agents (see Figure 1).



**Fig. 1:** User interest groups.

The model of user interests is organized hierarchically as an overlay over the domain ontology. Sub-areas in which the user has shown interest by issuing queries are represented, along with a value that indicates the strength of user interest in the area and a time stamp showing when the user made the last query in this area. The user's strength of interest in an area is calculated based on how many times the user has searched in this area, and how recently s/he has searched in this area. The user's interest in areas that are more general (higher in the ontology hierarchy) than the current area of search are also impacted, but in a much weaker way.

It is clear that in order to apply this approach, one condition is that all servants use a compatible representation of a domain ontology or ontology of services. There are various tools to developing such ontologies, e.g. DAML-S [4]. Another condition is that at two stages an association has to be made between areas/topics, keywords and resources /services: when the resource/service is introduced into the system, and when the user is making a query. The need to annotate a resource/service with respect to an area / topic and to provide keywords can be an impediment to both users bringing in resources/services from outside and to users searching, since it makes it necessary to make at least two more clicks (e.g. selecting the appropriate area /topic from a pull down menu) and entering keywords or selecting a directory to indicate the location of the file / service. To compensate for this extra effort we deploy a variety of motivation mechanisms discussed in the next section.

The model of the user's resources/services includes the files or services offered by the user (either actively, by bringing them in the system from outside, or by sharing files and services taken from others). When a new resource/service is created by the user, it has to be indexed by the user with respect to a certain area of interest and optional keywords, which are stored in an annotation file associated with the resource / service. When the user is offering a resource/service that has been taken from other users in the network, it does not need to be annotated. Each resource is associated with two measures of importance in the model – one denoting the subjective importance of the resource/service for the user, and one denoting the importance of the resource/service for the community.

The model of the user's relationships includes the users with whom the user has interacted frequently, i.e. from whom the user has downloaded files frequently and also users, who have downloaded files frequently from the user. These relationships are represented in a list where each relationship contains the unique id of the other user, the search area in the context of which the users have interacted, two numbers representing the strength and the balance of the relationship and a time-stamp

showing the last time the relationship was updated. The strength is a subjective factor reflecting how often the user has used resources/services from the other users, and how satisfied the user was with the interactions, i.e. if s/he kept the downloaded file or deleted it, if she used the file frequently. The balance of the relationship denotes the direction of services / files, i.e. who of the users predominantly uses and who offers resources / services. The next section explains how these three representations are created and updated.

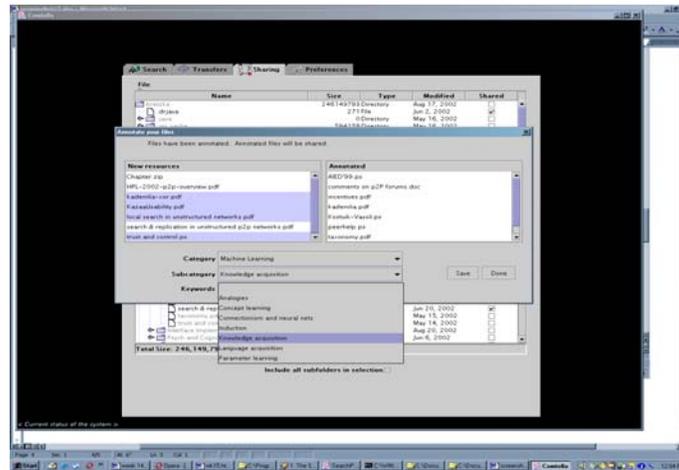


Fig. 2: The user interface for annotating a new resource. The black background visualizes the community and is replaced with a night sky image when there are other uses on-line.

### 3.2. Creating and updating the user model

The user model is updated each time the user adds a new resource or service to share, when the user searches services/resources in a particular area, when she downloads files or uses a service. There is also a possibility to update the user model by adding relationships with users who are searching for queries that are only passing through the server, i.e. to "listen" to the traffic and thus develop awareness about who looks for what. However, we are not exploring this option here, since it will lead to developing very large user models, containing relationships in all possible areas of search, which are not likely to be useful most of the time. We keep our model focused on the areas in which the user has been searching previously expecting that the user is likely to search again in these areas.

#### 3.2.1. Modelling user interests

Each server keeps track of the words / phrases entered by the user for search and adds the related areas to the model of user interests. The areas are retrieved from a lookup table reflecting the ontology of the domain. The strength of user interest  $S^a$  at time  $t$  in each sub-area  $a$  in the ontological hierarchy that is on the path leading to the

sub-area related to the query is updated according to a simple reinforcement learning formula:

$$S^a(e_t, t) = i * S^a(e_{t-1}, t-1) + (1 - i) * e_t \quad (1)$$

where the new (at time  $t$ ) evidence of interest  $e_t \in [0, 1]$  is calculated as  $e_t = 1/d$ , where  $d = 1 + \text{the distance}$  between the level of the sub-area of the query and the level of the area  $a$  in the ontology graph.

The parameter  $i \in [0.5, 1]$  is an inflation rate used to model the fact that older experiences become less important over time  $t$ , while the most recent experience is the most relevant (since the user's preferences may change over time). It can be fixed at a given value, say 0.5, giving equal weights to old and new evidence. The parameter  $i$  can also be a variable, which depends on the time elapsed since the last evidence of interest in this area, which allows capturing better the current tendency in user interests. An example is shown in Figure 3.

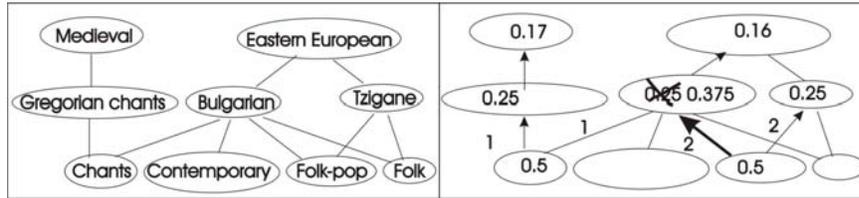


Fig. 3: Updating the model of user interests

### 3.2.2. Modelling the user resources / services.

To model the resources / services of the user, the server keeps track of the following:

- what files/service the user has downloaded/ used,
- what the user does with each downloaded file,
- what files/ services are downloaded and used from the user .

The actions of the user concerning the file / service (e.g. if s/he keeps or deletes the file, how often s/he accesses it or uses the service) are used to calculate the ranking of the file w.r.t. "subjective" importance for the user. The actions of other users concerning the file / service (e.g. how often it is requested / downloaded / used) are used to update the "objective" importance of the resource, i.e. how important is the resource for other users on a community level.

The ranking of resources is used by the server to compute the cost of keeping a given file locally. If the server has a strong relationship with another server in the group and the second server has the file, the first server has to decide, if it is worth keeping the file and eventually, suggest the user to delete it. More about the "specialization" of servers in a group is given in section 5.

### 3.2.3. Modelling the user's relationships

To model the relationships of the user, the server keeps track of the following:

- which servers respond with relatively many hits to the user's query,
- from which servers the user chooses to download files or to request service,
- the success of each download / service,

- who issues queries that result in hits in the list of resources of the user,
- who downloads files from the user.

The servents who have returned many hits along with those from whom the user has chosen to request service or download a file are entered in the user's list of formal "relationships" with an indication of the context of the particular area of interest (request) in which the relationship was created. The success of each download or service is used to update the *strength of the relationship* between the users using a formula similar to (1). Servents who are searching for files / services that are offered by the user and who choose to download files or use the services offered by the user are also added to the list of "relationships" of the user for the particular area of interest depending on the query used for the search.

Two users can be involved in relationships in several different contexts, indicating that they share interests in different areas. As a result, such users will have a very strong relationship (it will be the sum of the strength of the relationships for the individual areas).

The relationships that the user maintains with other users in a given area of interest can be sorted with respect to their strength. From these, a subset of currently active servents can be selected as the immediate neighbors to whom to send queries at any given moment. There are various policies for forming the neighborhood, which are described below.

One possibility is to select the top relationships from the area that corresponds to the current search area. This approach would follow the spirit of Ramanathan et al. [13] where the neighborhood changes with each search. However, since it is not sure that the user is going to continue searching in the area, or if she is going to issue another query in a different area, it probably doesn't make sense to change the whole neighborhood. Probably changing only a few of the neighbors, or forming the neighborhood of the global top  $n$  (across all areas) servents that are active at a given moment would allow better flexibility for future queries.

In addition to the relationship's strength and context, the servent keeps track of the balance (reciprocity) of the relationship. The servent of user X calculates the balance of its relationship with the servent of user Y as:

$$B^{XY} = N^{X \leftarrow Y} - N^{Y \leftarrow X} \quad (2)$$

i.e. the difference between the number of times when the user X has downloaded files from Y and the number of times when user Y has downloaded files from X. If the balance is negative, the user X "owes" user Y. The servent ranks the requests coming from other servents depending on the balance and the importance of the relationship. In this way, in a download queue, priority or more bandwidth will be given to request from important servents to the user, or to servents, from who the user has downloaded often resources or whose services were often used. Another "favor" that a servent X that "owes" to another servent Y does is not to decrement the time to live (TTL) i.e. the number of hops that the query can make, of a query sent by Y. In this way, the search horizon of a user who has contributed resources to users in the group increases.

The sum of the balances of all relationships of a user defines how much s/he has contributed to the community and how much s/he has consumed. Keeping a balance of each relationship allows maintaining a model of the user's contribution to individual users, to every interest group in which s/he participates and to the network

as a whole. It is used by the servent to provide an individualized motivational interface for the user, to persuade her/him to contribute to the community.

The described method for updating the user model uses an unobtrusive way of obtaining user information, posing no additional load on the user (such as requiring the user to enter ratings, or answer specific questions).

#### 3.2.4. Sharing user models among servents

While not involved actively in search, servents could communicate with each other and learn about other users with the similar interests. This learning could take two forms:

- *Direct*: without explicit request from the user, the servent sends queries in a particular area to find out servents that have resources and enters them in the list of relationships of the user. The strength of relationship can be updated as in [13] by the percentage of the number of hits by this servent over the total number of hits. In this way the servent explores the network. Also the time of request and reply can be recorded, to capture the compatibility in time patterns of being on-line. This approach, however, creates a lot of additional traffic in the network, and Gnutella has proven to be vulnerable to denial of service attacks caused unwittingly by servents that try to explore the network generating too much traffic [10].
- *Indirect*: by requesting from "friend-servents" the list of their relationships in a particular area. "Friend-servents" are those with whom the servent has a strong relationship in a given area. In this way the IDs of servents that have been frequently available and have provided a good service in a given area (e.g. good resources or services, did not interrupt the connection during download) are shared among the servents.

In a P2P network that is not very dynamic, i.e. most of the servents are active at approximately the same time, indirect learning will ultimately lead to all communicating agents from one interest group having the same list of relationships, which will lead to an implicit objective measure of quality / ranking of each servent within the group. This objective ranking will be contextualized, i.e. it will make sense only in the context of one interest group / coalition, since people behave differently in different communities. Of course, this will not prevent servents who have a high rank in one group to achieve a high rank in another group too.

There are a number of open questions concerning the interpretation of information received from others. One approach is when the servent requests relationships only from its "best friend" in an area.

However, it is also possible to request data from all existing "acquaintances" and use the strength of relationship with each source to compute the strength of relationship in the new "acquaintance". This approach is similar to approaches for trust propagation among agents in multi-agent communities [22,23]. There are different possible ways how information about a given servent coming along a chain of "acquaintances" can be interpreted: by voting among the sources, by averaging the strength values along the path, or by multiplying them. It is also necessary to define a policy for resolving conflicts between different chains of sources. Another question is if one representation of relationship strength is sufficient, or if each servent should keep two separate representations of the strength of relationship: a subjective one based on the servent's own experience, and objective reputation based on information

received from other servants. How can these two representations be combined and when?

We could argue against using global reputation measures in P2P systems because of the dynamics and the variety of preferences of users in the system, which can render any “objective” measures useless. For example, if user X requests a service from user Y always at 1 a.m. and never gets anything, since Y is never on-line at that time, X's strength of relationship with Y would be 0. However, Y could be a very active member of the community, providing useful files and services at other times. The strength of relationship that X has with Y reflects also the compatibility between X's and Y's preferences in the time pattern of usage, and it can not be generalized into one number without losing valuable context information. Therefore, just averaging the strength of relationship values of many users without considering the contextual information would not be appropriate. More sophisticated techniques than the currently existing reputation techniques deployed in Multi-Agent systems are necessary to retrieve information from appropriate servants, to interpret it in a context and purpose-dependent way.

#### 4. Motivating users to participate

Motivation of users to participate in the community is a crucial factor for the success for a P2P system. There is an ongoing discussion in the literature about the harm vs. benefit of free-riders for P2P systems [16]. The main argument in favor of free riding is that digital resources will never suffer from the tragedy of commons effect. They can be endlessly replicated, and each replica adds to the common (if the user who downloads the file shares it with the community). However, with free riders only the community can not create any wealth. There is a need for altruists or community-oriented users who create the resources or services once, before they start being shared and multiplied by free riders. Our experience with I-Help, a peer-to-peer system for help for over 2 years in the University of Saskatchewan shows that if the system lacks a "critical mass" of active users, it will never be able to take off [5, 11, 12].

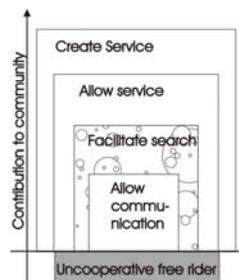


Fig 4: Levels of user cooperativeness.

In I-Help we observed several levels of user cooperative participation (Fig.4). Below we generalize them to the case of a P2P system where the servants deploy user

modeling as proposed in section 2. These levels of participation are characterized with decreasing degree of active user involvement (activeness):

- *create service*: creating new resources or services and offering them to the community,
- *allow service*: providing to the community disk space to store files for downloads or computing resources to enable a service that has been created by another participant in the community
- *facilitate search*: providing its list of relationships from the various groups in which the user participates to other users to facilitate search of files or services. This level of cooperativeness is possible if the servents model the "good" relationships with peers, as we proposed in the previous section.
- *allow communication*: forwarding ping-pong, query and hit messages, i.e. actively participating in the protocol of the network, i.e. participating in the peer infrastructure [2].
- *uncooperative free-rider*: downloading files or utilizing services when needed, but going off line immediately afterwards.

The "create service" level usually includes "allow service", "facilitate search" and "allow communication", i.e. it describes the most socially cooperative type of user behavior.

The more typical level is "allow service", describing a user who contributes passively to the community, by providing her resources and relationships, as well as the functionality of her servent to enhance the infrastructure of the community, but does not actively bring new resources or services into the system. As shown in [2], in Gnutella only a tiny minority falls into this category - 5% of the users is responsible for sharing over 70% of the files.

According to [2], the majority of users (66%) fall into the category, "allow communication" - they participate in the network infrastructure and therefore can be detected and taken into account. Unfortunately, there is no way to know at any moment how many users are "uncooperative free riders" or "creators of service" due to the lack of history in Gnutella and the anonymity, which does not allow to identify who first introduced a file into the system.

According to [20], more than 80% of the users of Mojo Nation were "1-time, 1-hour" users, and of the remaining users a significant part were "1 time, less than 24 hour" users. We observed a similar behavior of users in our I-Help system in certain classes, where the instructors failed to motivate a "critical mass" of active users in the beginning, when most of the users log-in just to try the system [18].

All popular file sharing P2P systems, like NAPSTER, Morpheus /KaZaA and LimeWire try to ensure both the "allow service" and "allow communication" levels of cooperation. Usually "allow service" is ensured by a default setting in the servent, which commands the downloaded files to be saved in a standard shared folder, so that other servents can find them. The "allow communication" level is achieved by making it hard to quit the servent. For example, a typical servent will not be quitted by clicking on the "close window" button, but will remain active on the task bar, until the user quits it once again explicitly. These default settings can be changed by the user, e.g. the downloaded files can be saved in a file different from the default shared Folder, or the servent can be quit by one click, but it requires more knowledge and

active involvement from the user, which is a form of "punishment" for uncooperative behaviour.

There are three principle ways of motivating users to participate in a community [17]:

- by trying to influence the user's feelings (of guilt, of belonging or owing to the group) to stimulate her altruism towards the community.
- by rewarding the user with visibility / reputation in the group depending on his/her contribution.
- by providing an economical model which ensures incentive for user contribution, (e.g. better quality of service, priority in the queues).

It is likely that choosing an appropriate way of motivation depends:

- On the personality of the user
  - On the nature or the user's interest in the area.
- Thus, the same user can be altruistic in one group, motivated by reputation in another group and by economic rewards in a third group.

#### 4.1. Motivating altruistic users

Altruistically motivated users are devoted to a particular cause (e.g. finding extra-terrestrial intelligence, cancer research or genome sequencing). They are likely to be active participants on the highest level (create service) in an interest group dedicated to the cause, like SETI@home. Influencing people to be altruistic for a given cause is a very difficult task; it requires a very detailed and broad model of user interests and of her acquaintances in the real world (who might be involved in a interest group with a certain altruistic purpose). This is still beyond the scope of the current user models and corresponding captology [5] (persuasion) techniques deployed in intelligent computer interfaces. A much more simplistic way that could hopefully influence the user is trying to provoke a feeling of quilt for not contributing to a community from which the user has taken a lot of resources. This could be attempted by using subtle cues like running messages in the window frame, or by a face or animal figure that changes its expression with the change in the owing balance of the user to the group (see section 3.2.3). We are currently developing a simple iconic avatar that represents the user in the community, and changes gradually to reflect the level of cooperativeness of the user. This level is computed from the sum of balances of the user's relationships with the members of the community, the number of files shared by the user, the relative duration in which the user's servent is active, and the number of user actions that are deemed as uncooperative, such as removing downloaded files from the shared folder, interrupting an ongoing file transfer or a service.

For each avatar there is a set of variants that differ in the level of friendliness of expression. Depending on the user's level of participation in the community, the avatar changes from a friendly sympathetic expression to an unfriendly and even vicious ghostly expression. This is accompanied with a running message on the bottom of the window suggesting what the user can do to participate more actively in the community, depending on the current level of participation of the user. The idea is that, similar to Oscar Wilde's "The Picture of Dorian Grey" [21], the user will be cued to reflect on her social behavior and how she can possibly change it for the better.

#### 4.2. Motivating reputation-aware users

Users motivated by social reputation are more likely to be active participants in groups where they already know some participants (even if by alias) and are known themselves. This impact on the user can be achieved through an appropriately designed interface of the servent that creates a global view of the group, visualizing in an appropriate way the servents that contribute most. We are currently developing a dynamic background image of the servent for this purpose (see Fig.2). It is inspired by the idea of a night sky where servents are represented as stars varying in size and brightness. The size denotes the amount of resources / services shared by the servent and the brightness denotes the amount of relationships that the servent has with other servents. The user can access an annotated version of the image, and by positioning the mouse on a star the user can see the name / alias of the servent. The star representing the servent of the user shines in a different color, so the user can see his/her significance in the community by the size and brightness. The image is generated / refreshed periodically to reflect changes in the group.

#### 4.3. Rewarding participation

Several P2P systems (most prominently, Mojo Nation [20]) rely on an economic model based on micro-payments to stimulate and reward participation. The basic assumption in the design of an economic model is that the effort and time spent bring new resources or services in the community have inherent costs. To take these costs into account, the resources/services should be made tradable. Thus the payment in a virtual currency (e-cash, mojo) may motivate a user to create new resources / services. It was shown theoretically [6] that if users/nodes are viewed as rational game-players, micro-payments for resources create a mechanism to balance the supply and demand of resources/ services.

Introducing an economy allows taking into account the different quality of resources or services provided. For example, a servent can jump to the top of the queue for a given service that is in a great demand at the moment, depending on the amount of mojo it is willing to pay for the service. In this way users who have contributed to the community and earned a lot of mojo are able to gain a better quality of service, which is a significant reward. Another way of reward with a better Quality of Service (QoS) within the system would be by protecting the user from unwanted advertisements.

A new way of providing better QoS that has not been proposed so far is to allow "richer" servents to "buy" themselves a wider search horizon by negotiating the TTL of each query. Thus users who have contributed have a better chance to find resources / files.

However, peer economies have raised a lot of criticism [16], mainly related to the fact that users prefer to pay flat rates rather than to bear the cognitive load of making decisions about micro-payment at each transaction. While this has been pointed as the reason for the failure of Mojo Nation to attract users, we believe, that the problem of the higher cognitive load can be avoided if the servent makes decisions on user behalf [19]. The user doesn't even need to be aware of the micro-payments happening in the

background; what she sees is a better or worse QoS depending on the economic state of her servent, which results directly from the cooperativeness of the user. In this way, if the accumulated currency is "cashed" in better QoS, it is important to ensure a gradual improvement or decrease in the QoS depending on the level of user contribution (level of accumulated currency) and always to maintain clear cues in the interface as to what is the reason for the increase / decrease in the QoS and what the user should do in order to improve it.

Motivating users to participate is very similar to teaching them how to behave as good citizens. A basic principle of good teaching is to provide a plenty of positive feedback, i.e. it is important to reward users for good behavior and not to give them the feeling that they are "punished" for bad behavior (at least not in the beginning), since they may withdraw entirely from the system. Of course, negative feedback should be present too, in carefully selected doses depending on the user's level of participation, so that the QoS doesn't deteriorate completely even for uncooperative free riders, since there is always the hope that they may become cooperative when they find the right interest group where they feel safe and comfortable.

It is possible to reward the accumulated currency outside of the P2P system. The choice of reward depends on the type of the interest group. One possibility that was adopted in Mojo Nation is to cash the currency in gift certificates from real-world vendors (possibly using the P2P system for advertisements). Another possibility, suggested in [16] is to give the benefits for cooperation up-front, in terms of for example, 100\$ off the next purchase of computer, or payment of the user's Internet service bill.

An economic model in a P2P system requires additional reasoning capabilities (utility computation and decision making) on behalf of the servents, it reduces the anonymity in the system [12] and requires centralized components to be introduced to be responsible for the currency / payment (as in Mojo Nation).

Our current implementation does not deploy an economic model. However, the advantages of such model as a motivation and regulation mechanism are significant and we will probably incorporate such a model in the future.

## **5. Group Evolution, Servent Specialization**

To achieve a system that is able to dynamically self-organize to optimize the distribution of resources, the servents should be equipped with reasoning mechanisms that allow them to make decisions about how to optimize the amount of resources / servers kept locally and the relationships it keeps with other servents. The reasoning of the servent should be motivated by individual benefit, i.e. like a rational agent, the server should optimize its individual utility function. However, on a global level, this individual behavior should result in a self-regulation mechanism, similar to the document routing model used by FreeNet that optimizes the efficiency of search and storage management for the whole community by locating files at particular servents.

Servents can specialize in two main ways:

- Across areas, servents can either specialize in a given group (interest area) or specialize as "hubs" maintaining relationships to well connected servents in many different areas,
- Within one area, with respect to providing either resources/ services or relationships ("hub"), or both.

Some servents can specialize in particular areas of interest by keeping mainly resources/services from this area and relationships to other servents in this area. Other servents will become "hubs", maintaining many relationships with specialized servents in various areas.

It is in the best interest for all servents to be members of all possible interest groups since in this way they will have access to a maximum number of good peers and their resources. Since the servents are able to learn from each other and share relationships, as described, this is technically possible. However, it too expensive for one servent to be a member of all coalitions in terms of disk-space for storing resources and user models, bandwidth, and computing power to reason about ranking and balancing relationships. It should be preferable for a servent not to specialize in particular area, if it has a strong relationship with an agent who is specialized in this area or an agent who is well connected with various groups. The decision in which group to specialize should take into account how often the user makes requests in the area of interest, the duration the user's interest in a given area as well as how many strong relationships the servent has with other servents in this area.

An indication for a persistent user interest in a given area is what the user does with downloaded files or how s/he organizes her services. For example, if the user creates a special sub-directory for downloaded files from this area, this is evidence that s/he has a long-term interest in the area. If the user deletes all the files related to a given area, this could be evidence that the user is no longer interested. Of course, the user might also want to be uncooperative and to avoid sharing the files, if she has copied them in another directory. Therefore, a fine grain analysis of users' actions not only within the P2P application, but also on an operating systems level, especially those related to file management are relevant.

If the servent has several strong relationships in an area (denoting similarities in users' tastes and time patterns) and the user does not frequently search items in the area, it may be better not to specialize in the area but only keep the relationships, in case the user wants to search again in the area. Of course, there will be no guarantee that these relationships will be available at a later time, but it is still a better starting point than the uninformed broadcast currently deployed in Gnutella.

Finally, the decision about specialization in an area of interest should probably involve the user. If this decision is not required too often, it might be appropriate to alert the user and request indication if she is interested in the area and would like the servent to specialize in this area, thus implying that the user participates in the community of other users interested in the area.

There are two kinds of specialization that a servent can choose within a given area:

- to keep a lot of resources or services;
- to serve as a directory service or a relationship hub for the group.

If we do not consider the costs of keeping resources or services [16], the benefit in keeping a lot of resources is for servents of users who access the resources frequently themselves. The benefit of keeping many relationships is that "knowing" other

servents that are either well-connected or well equipped with resources minimizes the number of hops a query will make and maximizes the likelihood that it will get a hit.

Even if we do not want to adopt an economic incentive model involving real payments (like the one used in Mojo Nation) the fact that keeping resources or services implies costs in terms of disk space and bandwidth has to be considered. Therefore, there is a motivation for the server to keep resources / services only if:

- the user frequently needs the resource herself, it can be retrieved locally.
- there is a reward that can be received for each download of the resource or usage of the service by other servents.

This reward does not need to be necessarily payment, but it can be, for example, a better quality of service. In order to measure this, some form of bookkeeping is necessary. We have argued in section 4.3 that a micro-payment computed and carried out by the servent, in a system-level economy, which the user doesn't even need to know about is necessary.

In this case, the servent first faces the question how to determine the price that it can charge other servents for the resource. The price depends on the cost of keeping the file/service (e.g. file size / hard disk space) and on the cost of providing the file for others (e.g. file size / available bandwidth). It also depends on the current demand for the resource/service, which can be measured by the number of requests queued. If the cost component of the price is too high as a result of small hard disk and/or bandwidth and there is no demand (either from the user or from other servents), and there are good relationships in the group of interest to servents that have the file, there is probably no good reason to keep the file. The servent should suggest to the user deleting the file from the shared folder. It is important to let the user make the ultimate decision because if she feels lack of control, she may distrust the servent and the whole P2P system [16].

In an economic model, users with large storage and bandwidth capacity can afford to specialize as servers. In this way they will earn currency with each download, while other users may become pariahs, i.e. go in "depth" and create very unbalanced relationships. These users will have to contribute resources or services to the community or do a lot of searching in different areas to allow their servents to build relationships and become specialized as relationship hubs.

The motivation for relationship hubs to specialize in this way in an economic model is that they can "charge" a minimum connection fee for each query passed through them that results in hit.

We are currently investigating what could be a simple and effective pricing policy for micro-payments. Our next steps will be simulating the economy to see what parameter values can be controlled to achieve an equilibrium state and implementing the decision making process for servent specialization.

## 6. Discussion and Conclusions

We propose that the servent keeps a user model, which allows it to know the areas in which the user has interests and who are "the friends" of the user in each area of interest. Unlike the approach proposed in [13], where the interest groups of users were

highly dynamic and changed rapidly to reflect the current search performed by the user, our approach relies on the assumption that users have long-term interests and are likely to search repeatedly in the same area at different times. Therefore, it makes sense to keep track of all "interest groups" of the user, to be able to use them again when a new search happens.

In addition, our approach assumes that the topology of the network in an established active interest group does not change too rapidly. We believe that even though one of the biggest strengths of P2P systems is the ability to work in a highly dynamic environment, where servants (e.g. users) can come on-line and leave at any time, there is a pattern of behavior that can be tracked (locally, by the individual servants) and adapted to, for the benefit of the users. The strength of the relationship between two users reflects not only a certain similarity in tastes and interest, but also a compatible pattern of being on-line. Users, who are related with strong relationships, who have been able to share valuable files / services in a mutually convenient time in the past, are likely to be able to do this again in the future.

The idea of flexibly changing the horizon for search has been proposed in [13], depending on the how promising is the immediate neighbor to whom the request is sent. We extend this idea with the possibility of negotiating the search horizon between the servants considering the strength and the balance of relationship with the servant, thus giving advantage to servants who have been cooperative in the past.

We also discussed various ways to motivate users to contribute actively in the community. Unlike [2], we don't see free riders as a necessary evil, but we don't think they should be praised either (unlike [16]), since we realize that it is impossible to build a successful P2P sharing community with free riders only. In contrast with [13], where goal is to reduce network traffic, and with [6], where the goal is to pay users for sharing files, our goal is simply to ensure a better quality of service to users who contribute to the community.

Our future steps and evaluating the advantages and disadvantages of the proposed design of COMUTELLA with respect to user satisfaction and performance.

**Acknowledgements.** The author thanks Christopher Cox, who implemented the Comutella servant, and Helen Bretzke, who implemented the interface and developed some of the motivational strategies. Yamini Upadrashta implemented the user model. Thanks go also to Ralph Deters, John Kaufmann, Yao Wang, Chris Brooks. This work has been funded by two NSERC undergraduate summer scholarships, the CRA-Women peer mentor program and the NSERC operating research grant of the author.

## References

1. Adamic (2000) The Small World Web. Technical Report, Xerox PARC.
2. Adar E., Huberman B. (2000) Free Riding on Gnutella. First Monday, vol. 5, no. 10. Also available on line at: [http://www.firstmonday.dk/issues/issue5\\_10/adar/](http://www.firstmonday.dk/issues/issue5_10/adar/)
3. Cornelli, F., Damiani, E., De Capitani di Vimercati, S., Paraboschi S., Samarati P. (2001). Implementing a Reputation-Aware Gnutella Servent. Proc. Int. Workshop on Peer-to-Peer Computing, collocated with Networking 2002, May 19-24, Pisa, Italy.
4. DAML Services: <http://www.daml.org/services/>
5. Fogg B.J. (1998) Persuasive Computing: Perspectives and Research Directions, Proceedings CHI'88, [http://hci.stanford.edu/captology/Key\\_Concepts/Papers/papers.html](http://hci.stanford.edu/captology/Key_Concepts/Papers/papers.html)

6. Golle Ph., Leyton-Brown K., Mironov I. (2001) Incentives for Sharing in Peer-to-Peer Networks. Proceedings EC'01, October 12-17, 2001, Tampa, Florida, ACM press, 264-267.
7. Greer J., McCalla G., Vassileva J., Deters R., Bull S., Kettel L. (2001) Lessons Learned in Deploying a Multi-Agent Learning Support System: The I-Help Experience, Proceedings of AI in Education AIED'2001, San Antonio, IOS Press: Amsterdam, 410-421. Available on line at: <http://julita.usask.ca/homepage/public.htm>
8. Joseph S. (2001) NeuroGrid: Semantically Routing Queries in Peer-to-Peer Networks. Proc. Int. Workshop on Peer-to-Peer Computing, collocated with Networking 2002, May 19-24, Pisa, Italy, available on line at: <http://www.elet.polimi.it/p2p/>.
9. Keleher P. Bhattacharjee S. Silaghi B. (2002) Are Virtualized Overlay Networks Too Much of a Good Thing? in Electronic Proceedings for the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02) available at: <http://www.cs.rice.edu/Conferences/IPTPS02/>
10. Manjoo F. (2002) Gnutella Bandwidth Bandits. Salon, August 2002, available on line at: [http://www.salon.com/tech/feature/2002/08/08/gnutella\\_developers/print.html](http://www.salon.com/tech/feature/2002/08/08/gnutella_developers/print.html)
11. Metcalfe's law: available on line at the Management Issues website: <http://www.mgt.smsu.edu/mgt487/mgtissue/newstrat/metcalfe.htm>
12. Milojicic D., Kalogeraki V., Lukose R., Nagaraja K., Pruyne J., Richard B., Rollins S., Xu Z. (2002) Peer to Peer Computing. Technical Report HPL-2002-57, HP Laboratories Palo Alto.
13. Ramanathan M. K., Kalogeraki, V. Pruyne J. (2001) Finding Good Peers in Peer-to-Peer Networks. Technical Report HPL-2001-271, HP Laboratories Palo Alto.
14. Reed D. P. (1999) That Sneaky Exponential-Beyond Metcalfe's Law to the Power of Community Building. Context magazine. Spring 1999. available on line at: <http://www.contextmag.com/setFrameRedirect.asp?src=/archives/200208/0ToC.asp>
15. Reed D.P. (1999) Weapon of Math Destruction, Context Magazine, Spring 1999. at: <http://www.contextmag.com/setFrameRedirect.asp?src=/archives/199903/DIGITALSTRATEGY.asp>
16. Shirky C. (2000) In Praise of Freeloaders, The O'Reilly Network. Available on line at: [http://www.oreillynet.com/pub/a/p2p/2000/12/01/shirky\\_freeloading.html](http://www.oreillynet.com/pub/a/p2p/2000/12/01/shirky_freeloading.html)
17. Vassileva J. (2002) Motivating Participation in Virtual Communities, Proceedings of the 12th International Conference of Women in Engineering and Science, ICWES'12, July 27-31, 2002, Ottawa, Canada.
18. Vassileva J., Deters R. (2001) Lessons from Deploying I-Help, in J. Whatley & M. Beers (eds.) Proceedings of the Workshop on Agents and Internet Learning, AIL'2001 at the Autonomous Agents'2001 Conference, Montreal, May 28, 2001.
19. Vassileva J., J. Greer, G. McCalla, R. Deters, D. Zapata, C. Mudgal, S. Grant (1999) A Multi-Agent Approach to the Design of Peer-Help Environments, in Proceedings of AIED'99, Le Mans, France, July, 1999, 38-45. Available on line at: <http://julita.usask.ca/homepage/public.htm>
20. Wilcox-O'Hearn B. (2002) Experiences Deploying a Large-Scale Emergent Network, in Electronic Proceedings for the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02) available on line at: <http://www.cs.rice.edu/Conferences/IPTPS02/>
21. Wilde O. (1891) The Picture of Dorian Gray, available on line: <http://www.bibliomania.com/0/0/57/103/frameset.html>
22. Yu, B., Singh, M. (2002) An Evidential Model of Distributed Reputation Management, Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: part 1, July 2002, Bologna, July 15-19, 2002.
23. Yu, B., Singh. M. (2002) Emergence of Agent-Based Referral Networks. Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: part 3, July 2002, Bologna, July 15-19, 2002.