Organizing Service-Oriented Peer Collaborations

Asif Akram and Omer F. Rana

Department of Computer Science
Cardiff University, UK
{A.Akram, O.F.Rana}@cs.cf.ac.uk

Abstract. Locating suitable resources within a Peer-2-Peer (P2P) system is a computationally intensive process, with no guarantee of quality and suitability of the discovered resources. An alternative approach is to categorise peers based on the services they provide – leading to the interaction of peers with common goals to form societies/communities. Organization of peers in different communities is suggested to be useful for efficient resource discovery. The concept of communities is explored with reference to questions such as: why communities are desired? How they are formed? How communities work and interact? What are different possible types of communities and their overall behaviour? What are the advantages of community formation? The communities are adaptive in nature and evolve based on changes in their operating environment – such as changes in neighbouring communities. We suggest the benefit of this approach for resource discovery, and use a JXTA prototype to illustrate the concepts. The particular focus of this paper is to explore different types of organizational structures in the context of software provision in the context of service communities.

1 Introduction

Emerging distributed computing paradigms, such as Grid Computing [2], comprise of resources which may freely join and leave the system – and are said to constitute a "Virtual Organisation". Identifying how such an organization should be structured is an important part of developing more useful and efficient collaborations. Such Virtual Organizations already exist in both science and engineering projects – whereby a collection of scientists come together to solve a single large problem. In the High Energy Physics domain (such as the D0 project [15]), many groups and institutions come together for collaborative problem solving. Identifying a service-based infrastructure, which makes use of Grid technologies, is therefore important to support multi-disciplinary science in the future.

There is no reliable way to discover such dynamic peers and resources, making it impossible to have updated information about all available resources. However, without such information resource discovery becomes a time-consuming process and imposes an overhead on network access [1]. As the number of peers grow, the rate of possible interactions among peers increase exponentially. It is not scaleable to interact with all peers to discover appropriate resources, and all peers are unlikely to have information about all other peers. Restricting interaction within a set of peers is a key factor to scale the resource discovery problem. Peers can be categorised based on criteria such as the type of service, quality of service, etc. Any initial cost in categorising resources can provide benefits for discovering preferable resources

M.E. Orlowska et al. (Eds.): ICSOC 2003, LNCS 2910, pp. 451–466, 2003.

without a large discovery cost subsequently – thereby leading to the development of "communities". A community of autonomous peers, or community of communities, or even hybrid community, can exploit the scaling effects and benefit from the presence of other communities [3]. Further enhancements in the discovery of resources are possible if "similar-minded" communities i.e. communities offering similar services/resources or have similar resource requirements share their knowledge of a distributed environment [6]. Thus, the discovery problem of resources is scaled to known similar-minded communities where the probability of resource availability is likely to be high.

The concept of communities is very similar to interactions between different departments at a University. For instance, a lecturer can be a member of different faculties e.g. a mathematics lecturer teaching calculus to computer science students. This analogy helps us to define two terms, *Expertise* and *Interest* [4], [5]. Expertise of a peer is the basic service provided by that peer and Interest of a peer is the service/services provided by other peers which are supportive to its main service. In this example, a mathematics lecturer is an expert in mathematics and may not have any interest in computer science. If another department introduces a new calculus module, for instance, then instead of contacting all university lecturers (peers) individually it is preferable to contact the mathematics department (community) – as this improves the possibility of locating an appropriate peer capable of offering such a service. A common problem in Grid Computing is what Davis and Smith refer to as the "connection problem" [7], where peers need to find other suitable peers to co-operate with, assist, or interact with. "Focused Addressing" [8] is one solution to the connection problem where requests are sent to particular subset of peers, believed to assist the requesting peer. Communities in Grid Computing exploit the concept of societies. Similar-minded peers who either have similar expertise or interest in each other's expertise form societies. In societies or communities, the interests of individuals are protected, whilst allowing them to interact with each other for common benefits. It is a concept similar to the producer and consumer paradigm; if a producer does not market its service/s in a proper way, then the consumer may not be able to locate the producer.

The development of communities should allow similar-minded peers to be combined/grouped. Furthermore, the process of community formation should be automatic, and enable individual peers to benefit from joining one or more communities. One may assume each peer to be selfish, and only interested in the services they require and be in a position to be easily discovered by clients. If both conditions are not met then individual autonomous peers may not have any incentive in joining a community. Peers themselves are not loyal to communities but benefit from being in a community in accordance with their personal goals, thus creating a social network, which is self-supportive in nature.

2 Communities

Individual peers, although selfish, are expected to interact with each other in some way. Co-operation of one form or another therefore becomes essential. Each peer prefers to be in an environment where it may be easily discovered by a suitable client, and can locate other peers with minimum efforts, thus enhancing its utility. Utility of peer is its effectiveness for other community members, and activeness within the community. Hence the more useful a peer to the objectives of the group, and the more activities it performs, the higher will be its utility. We assume there is some incentive for each peer to be discovered by others – perhaps based on some pay-off (or reward). In a Grid system, this could be to enable a peer to maximize its resource utilization. Peers providing different services, even with different service attribute i.e. quality of service, reliability, etc may be grouped together based on attributes such as type of services, resources and domains [6]. Similar-minded peers are grouped together to form communities; these communities can be treated as autonomous units in Grid Computing. It is proposed that a Grid environment is a collection of autonomous communities, which are dynamic in nature, as peers may join and leave at any time. Different peers either providing similar services/resources or interested in any particular services/resources interact with each other to form communities. Peers collaborating with each other to form a community should have one special peer acting as a community manager – we call this the Service Peer. Each community has one Service Peer with dual responsibility of not only managing the member peers but also keeping track of other communities with which they interact on behalf of member peers. A Service Peer is similar to an ordinary peer with respect to service/s and resource/s but with few additional responsibilities, the concept is similar to peers in JXTA where a peer can have additional responsibility of rendezvous peer [11]. Interaction between communities to discover new resources/services is only through the Service Peers. Direct interactions between peers for discovery of resources are prevented to restrict message traffic across communities [9].

3 Community Formation

Each Service Peer is an empty community which is automatically created when Service Peer is created. Service Peer may also offer specialist services, such as file sharing, content management service etc. along with different management services (section 6). A Service Peer manages a membership policy based on its expertise, and one that restricts the entry of other peers into a community. If no such expertise is held, then the membership policy is decided by the expertise of the first peer that joins the community. A new peer first tries to discover the Service Peer which may have interest in its capabilities/services. If the interests of a Service Peer are different, the new peer is referred to other Service Peer/s, or the new peer tries to locate alternative Service Peer/s with compatible interests (if the contacted Service Peer has not responded). A Service Peer and all peers registered with it constitute a community. A Service Peer manages all peers within the community and interacts with neighbouring Service Peers on the behalf of its member peers. A Service Peer therefore encodes the combined capability offered by all peers within its community. A Service Peer is essential for the bootstrapping of a new peer, as it supports a new peer to discover enough network resources to sustain itself. We make the following assumptions, which may be treated as the limitations of the system, but these assumptions will be justified latter:

1. More than one community exists in the system at a time.
2. Neither every peer nor its Service Peer know about every other peer or Service Peer, nor does any peer or Service Peer require complete information about all other peers or Service Peers.
3. It is appropriate to group peers on the basis of common attributes [6] i.e. type of service, domain, quality of service, into communities or clusters. Any given peer might be in more than one community or cluster simultaneously, depending on the services it is providing.
4. Each community has expertise depending on the type of services offered by member peers, but it can have different interests from member peers. Expertise and interests are two different things and should be kept separate [4], [5].
5. Each peer is impartial to any other peer, and only interested in the services the other peer provides.
6. A Service Peer may share information with other peers with different expertise, depending upon the nature of its interest. It is not necessary for only Service Peers with common interest to communicate with each other.
7. It is possible to prioritise Service Peer interests and expertise, such that we can say that a particular interest or expertise of Service Peer A is more like Service Peer B than Service Peer C.
8. Each community has at least one Service Peer.
9. A community can be created in a way similar to a JXTA Group with certain expertise and interests without any member peer/s.

4 Type of Communities

Individual autonomous peers have expertise and interests in specific resource/s. Based on these expertise and interests, peers are grouped together, but expertise and interests are not the only criteria for categorizing peers. Communities/societies can be of different types i.e. Competing Communities and Co-Operative Communities. We outline aspects of these different types of communities, and how they help structure a P2P system.

4.1 Competing Community

In a Competing Community all peers have the same expertise and to some extent member peers are providing the same service/s – although some service attributes may vary. Similarity in expertise may develop competition amongst member peers, as member peers have to compete with each other to get selected by a client. The competition is mainly for attributes which are not shared by peers like service quality, cost and hardware resources available [6]. Overall, these types of communities will result in competition and improved quality of services within the community. This concept is similar to what happens in human societies. For instance, different hardware manufacturers advertising in the same news media for the same product. Such manufacturers should differentiate themselves in some way to be selected by a buyer. A Competing Community may have two types of Service Peers: (i) a *Service-Oriented Service Peer, and a* (ii) *Non Service-Oriented* Service Peer. A *Service-*

Oriented Service Peer manages all member peers for completion of any single request/service. If any one peer fails to complete the assigned task, then a *Service-Oriented Service Peer* can assign that responsibility to another available peer within the community (Figure 1a and 1b), and this change will be transparent to the client application.

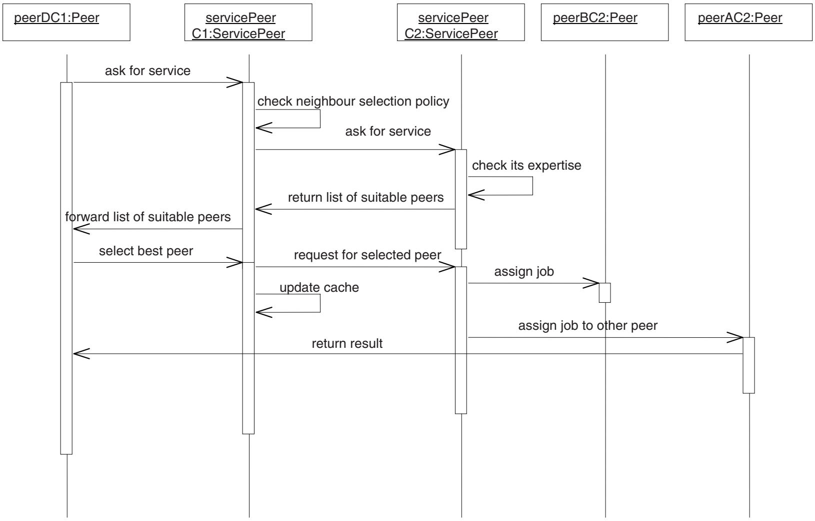

peerDC1:Peer
servicePeer C1:ServicePeer
servicePeer C2:ServicePeer
peerBC2:Peer
peerAC2:Peer
ask for service
check neighbour selection policy
ask for service
check its expertise
return list of suitable peers
forward list of suitable peers
select best peer
request for selected peer
assign job
update cache
assign job to other peer
return result

Fig. 1a. Sequence Diagram for Service-Oriented Service Peer in Competing Community

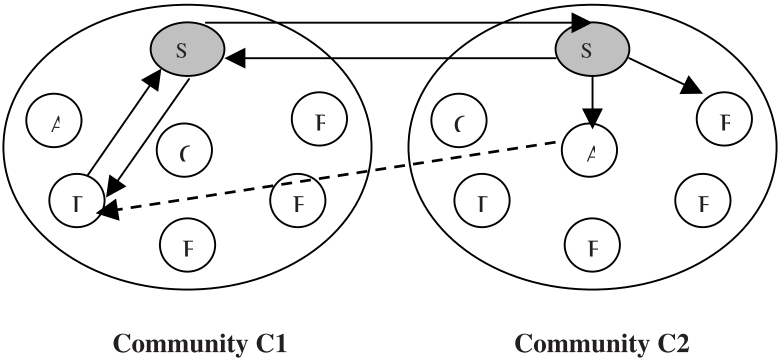


Fig. 1b. Peer A of C2 completing the task on failure of peer B of C2 in Service Oriented Competing Community

A Non Service-Oriented Service Peer in a competing community will not interfere with a client application and the service provider peer. From Figure 1, a Service Peer of community C2 informs Service Peer of community C1 about the unavailability of peer B.C2. The Service Peer of community C1 must now contact another community (such as C3) for the completion of the request, while peer D.C1 will remain unaware of the whole process of discovery and will only receive the new list of potential peers for selection. Figure 2 illustrates the failure of one peer in a Competing Community with *Non Service-Oriented* Service Peer.

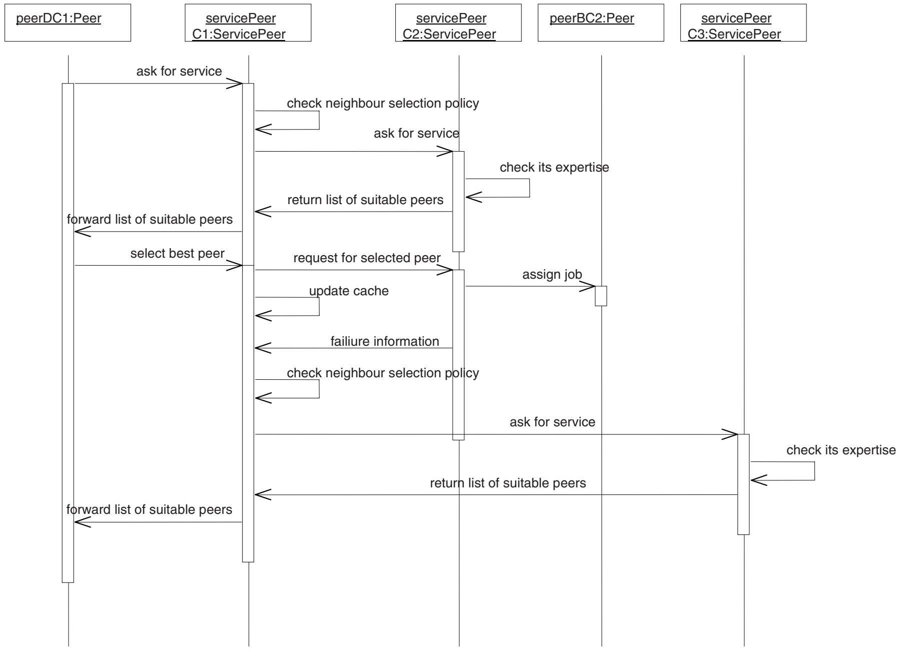

peerDC1:Peer
servicePeer C1:ServicePeer
servicePeer C2:ServicePeer
peerBC2:Peer
servicePeer C3:ServicePeer
ask for service
check neighbour selection policy
ask for service
check its expertise
return list of suitable peers
forward list of suitable peers
select best peer
request for selected peer
assign job
update cache
failiure information
check neighbour selection policy
ask for service
check its expertise
return list of suitable peers
forward list of suitable peers

Fig. 2. Sequence Diagram for Non Service-Oriented Service Peer in Competing Community

Regardless of the type of Service Peer an individual community has, the overall result is less resource consumption in discovery for new resources. A Service Peer utilises a neighbour selection policy based on the expertise and interests [6] of other Service Peers it interacts with. It is more efficient for a Service Peer to maintain a neighbour selection policy instead of individual peers in the community, as a single policy is applicable to all peers, each member peer can be restricted by that single policy and each member peer can also benefit from the previous interactions of a Service Peer.

4.2 Co-operative Community

In Co-Operative communities all peers provide different services i.e. have different expertise, but have interests in the expertise of member peers. Each peer within such a community is providing a limited set of services, which may not be utilised individually, but along with services and resources of other member peers within the same community. As an analogy to electronic markets, a motherboard manufacturer needs other manufacturers to supply CPU, Hard Disk, Memory Card and VGA Card etc to be viable. In such communities, each peer is dependent on at least one other member peer which may be dependent on the service of any third member peer. Hence, when one peer is selected by a client, then there is a better possibility of selecting another member peer. This mutual co-operation is suitable for those peers which provide very simple and basic services, thus each peer directly or indirectly support the service of other member peers in the Co-Operative Community.

A Co-Operative Community has a few advantages, as a single community is providing the complete service using different member peers. Hence, a client may not have to discover different resources for accomplishing a single task - which means efficient discovery and less interaction with different communities. The effectiveness of Co-Operating communities is however dependent on the co-ordination of individual peers.

A Co-Operative community may also have a: (i) a *Service-Oriented Service Peer (SOSP),* and a (ii) *Non Service-Oriented* Service Peer *(NSOSP).* A service provided by a Co-Operative Community is divided into different independent phases which must be co-ordinates into a workflow. Each individual member peer works on a particular phase of the client application, and returns its results to either a Service-Oriented Service Peer or directly to another member peer responsible for the next phase. In the case of a Non Service-Oriented Service Peer, this does not make any difference as long as interactions are within the community. Involvement of a Service Peer *(Service Peer Oriented Community)* after each phase is essential when different peers can accomplish a particular phase to enable selection of the most appropriate peer within the community. To reduce traffic within a community, each member peer may know about the sequence of phases and peers involved, and only transfer the final outcome to the Service Peer, as illustrated in the scenario in Figure 3.

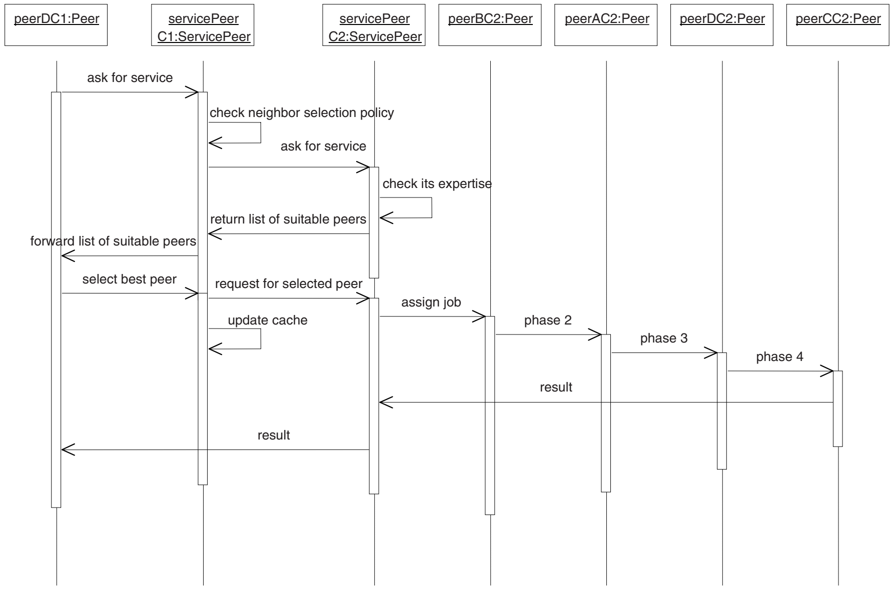


Fig. 3. Sequence Diagram for Service-Oriented Service Peer in Co-Operative Community

4.3 Goal Oriented Community

Another type of community is one which has a collection of peers to achieve a particular goal. In this community, a Service Peer is required to accomplish a user-defined goal related to its expertise. Based on its goal function, a Service Peer searches for ordinary peers and on locating appropriate peers it invites them for limited membership. Membership in such a community is only allowed to accomplish the assigned task – and membership of the community is terminated on task completion. These communities are a strict type of Co-Operative Communities, in which peers interact with each other in a pre-defined sequence dependent on the context of the service/s. Goal-oriented communities may also be important in self-organising systems, whereby the interaction between member peers is not pre-defined, but the services required are. In such instances, member peers may interact with each other in arbitrary ways to achieve a given end result.

4.4 Ad Hoc Community

Here, peers may be in different communities regardless of the nature of those communities, but still work together as a team. In ad hoc communities peers interact directly with each other without interference and involvement of a Service Peer. Two peers belonging to different communities providing two different but supporting services form the basis of an ad hoc community, as long as both concerned communities have agreed to use each other’s service. This is an exceptional community in which involvement of Service Peer is very limited and member peers in ad hoc communities are more independent.

4.5 Domain-Oriented Community

Such a community is formed by linking together similar-minded organisations and institutions, instead of the services they provide, such as academic communities, research communities, and open-source communities. Hence these communities are domain-oriented rather than service-oriented. A typical Domain-Oriented Community is quite diverse in nature and cannot be categorised on the basis of the provided services. Furthermore, peers in these communities may not have interest in each other. Communities such as these can be restricted to a particular geographical location, specific organisations etc. The importance of such communities is that they enable common mechanisms to view common problems that a given community is likely to encounter. It is possible for members of the community to solve the same types of problems in common ways – using different types of services.

5 Architecture of a Community

The architecture of communities should be simple and supportive to the main purpose of their formation. Each type of community has a similar architecture, with one Service Peer, which manages the whole community. A Service Peer is similar to an ordinary peer with respect to the services and resources it provides, but with additional management services (such as a rendezvous peer in JXTA [11]). Each peer must be a member of one or more communities.

5.1 Discovery and Membership

A newly created peer tries to locate Service Peers in a Grid environment by using a Service Peer discovery message sent on the network (provided the local network supports broadcasting or multicasting). Alternatively, a newly created peer may only send a message to one Service Peer, which then either responds directly or refers the message onwards. If a peer is interested in joining a particular community, it sends a request to the Service Peer for that community. It is not necessary that each Service Peer which receives the discovery message will reply, and even a reply does not imply the acceptance of membership. A peer will act solely to maximize its long-term utility, so during membership of particular community, peer will take discounted estimate of future rewards into considerations [9] (to decide whether to remain in the community).

5.2 Individual and Collective Interests

Each member peer has a unique interest which varies from peer to peer. This interest is based on the type of service(s) that the peer provides. A Service Peer is responsible for intercommunity interactions, and must reflect the interests of all member peers within a community. A Service Peer must therefore summarise the interests of a community by combining interests of each of its members into a list. This is currently achieved by listing all the services that are being offered by each member peer within the community.

5.3 Internal Rating of Members

A Service Peer rates member peers according to their activeness (number of times a successful service has been provided), the type(s) of expertise they have, the quality of service they offer, etc. This internal rating mechanism is mostly general, but each Service Peer may have different criteria based on its local policies. A Service Peer selects the best available member peer/s as a result of a request for any service from a client, based on the expertise provided by the community. On selection of any member peer by a client, the Service Peer will increment the internal rating of the member peer. The most active peers will have the maximum rating; if any member peer is unavailable, overloaded or not responding for a long time, then the Service Peer will not select it for subsequent recommendations. A similar rating mechanism is used for Service Peers of external communities, and used in supporting query referral [5]. Regardless of the type of community, each peer aims to maximise its rating over a particular period. Non-availability of service/s from any peer will also affect the overall ranking of the peer.

5.4 Multiple Memberships

Individual peers can be members of different communities, and the selection and membership of a community is based on the expertise and different interests of each individual peer. Membership for a new peer is completely dependent upon the type of service offered, organizational domain of peer, quality and completeness of service and finally the expertise and interest of community. As the internal rating of the peer increases, it will have a better chance to move to other communities where its services are likely to be in a higher demand. The internal rating provided by a Service Peer may also be a criterion that a particular peer may wish to make known; hence, each peer tries to be active in all communities for which it has valid membership. Membership of many communities can drastically affect the internal ratings of a peer, as a peer has limited resources. However, if a particular community does not get a large number of requests, then the services offered by a member peer may not get utilized properly. Consequently, in communities where the number of requests is not that significant, it is in the interest of a service providing peer to belong to multiple simultaneous communities to increase its utilization. Membership policy for each joined community will be different based on the type and expertise of that particular community.

5.5 External Ratings of Communities

Service Peers maintain information of member peers and a restricted set of other communities; this interest is governed by the expertise and interest of each Service Peer. Based on interactions with other communities, a Service Peer records external ratings of other Service Peers, and this rating is considered during future interactions [5]. Each community has its own rating of other communities as each one has its own expertise and interests. The higher the rating, the more compatible it is with other community [10]. The rating of communities can be used as a measure to identify communities which are likely to be more effective when working together. Based on a policy, a Service Peer may not reveal such rating to other Service Peers.

5.6 Virtual Community of Communities

It is desired that each community should have a list of characteristics i.e. expertise, interest of other communities known to it. A Service Peer will try to match the characteristics of different communities known to it, and the characteristics it has about itself, to form a virtual community of communities [5]. This virtual community of communities is based on the perspective of the Service Peer, as other communities will have their own virtual community of communities with entirely different participating communities [4]. For instance, a Service Peer of community A has a Service Peer of community B in its list of Service Peers (i.e. virtual community of communities) based on similar, but it is not necessary that Service Peer of community A is also in the virtual community of Service Peer of community B. Service Peer of community A and B have some common characteristics, and these may be important for Service Peer of community A but not for Service Peer of community B. The nature of these communities change over time based on the changes in their membership. Neighbor selection [10] is influenced by the commonalities in characteristics i.e. expertise and interests and external rating of the community.

5.7 Information Sharing

Service Peers frequently exchange their contents with other communities in its virtual community of communities, but there is no guarantee that two Service Peers from different communities have the same view. Such content may also include ratings of other communities, thereby indicating the suitability of such external communities to host particular types of resources/services. As each community has different neighbors due to its different expertise and interests, this exchange of information may help identify many other communities which may provide useful expertise, to varying extents of usefulness. Normally, a Service Peer will interact with other Service Peers (communities) which have similar expertise and interests. A Service Peer updates its virtual community based on the (dynamic) ratings of other Service Peers – and records this for a lease duration – a limited period after which it tries to re-build its acquaintance list. Each Service Peer maintains a list which contains the names and contents from the last *r* Service Peers (communities) that this Service Peer has communicated with [4], [6].

5.8 Learning and Adaptivity

Peers are free to join and leave communities, and may join different communities of different types. This makes tracking of peers a difficult job for the Service Peer. A peer will act solely to maximize its long-term utility during membership [14]. After a certain time period each peer will primarily aim to be in a community or communities where it has maximum rewards. The system reaches equilibrium over time, provided the environment does not change significantly, and the internal and external rankings of the communities will not change significantly as new peers are added. Achievement of stability in the system is an important objective in the long run. However, this is also dependant on the rate of change of the environment within which the communities exist. A dynamic operating environment is likely to prevent the system stabilising, as new members may be added/removed from the system rapidly, and the services offered by the members may also change in unpredictable ways. Communities are much more consistent in nature. They provide this consistency by giving membership only to “similar-minded” peers – based on their expertise and interests.

6 Common Services in Communities

Each community requires a set of common services to function adequately. These common services generally offer management capability to enable individual members of a community to function well. Communities require different services to manage the interests of individual autonomous peers.

Different types of communities requires different common services, and these may be offered by a Service Peer i.e. Application Server with specialist middleware, or separate peers providing management services i.e. Application Server using middleware from different vendors [12]. A Service Peer works as a gatekeeper and manages all services residing on different peers within a community, and every message from/to another community passes through it (except in the case of an Ad Hoc community). Communities must support one or more of the following services:

1. **Security Manager Service:** Focuses on the requirements for supporting authentication, authorization, accounting, and auditing of access to and services provided by the community.

2. **Scheduling Service:** Schedules responsibilities to different peers, and monitors different phases of job execution.
3. **Transaction Manager Service:** Ensures the dynamic (or static) load balancing within the community to maximize throughput when required.
4. **Concurrency Controller Service:** Co-ordinates two or more peers providing the same service to the same client.
5. **Resource Monitoring Service:** Monitors use of internal resources among member peers within the community and external network resources for inter community interactions. Such monitoring may be supported through specialist tools that are available on hosting platforms for particular peers.
6. **Performance Controller Service:** Responsible for finding non-overloaded peers to maintain a given Quality of Service. Hence, monitors the performance and activity of internal resources for better external rating of the community. May work in liaison with the Resource Monitoring Service.
7. **Policy Manager Service:** Implements the policy for a specific type of community i.e. membership policy, neighbour selection policy, internal rating policy, inter-community interaction policy.
8. **Networked Information Discovery and Referral Services:** Manages the external rating, availability, quality and expertise of neighbouring communities and discovers communities of interest.

Different communities' offers different services and Table 1 shows the essential services offered by different communities. It is clear from Table 1 that each community doesn't have all services.

Table 1. Comparison among different communities based on type of components required

Community Type	Security Manager	Scheduler	Transaction Manager	Concurrency Controller	Resource Monitor	Performance Controller	Policy Implementer
Co-operative NSO	X		X			X	X
Co-operative SO	X	X	X		X		
Competing SO	X	X	X	X	X	X	X
Competing NSO	X				X		X
Goal Oriented	X	X	X		X		X
Ad Hoc	X					X	
Domain Oriented	X				X		X

7 Comparison of Different Communities

Each type of community has advantages or disadvantages and there is no easy way of comparing them and requires different components for optimized performance. Below is the comparison of different types of communities based on factors like resources overhead, reliability etc. Table 2 compares different communities with respect to use of different additional resources.

Table 2. Comparison of different communities based on required components

Community Type	External Interactions	Internal Interactions	Network Resources Required	No. of Components	Efficient Resource Usage	Service Replication	Reliability
Co-operative NSO	Min	Avg.	Min	Avg.	Max	Avg.	Avg.
Co-operative SO	Min	Max	Min	Max.	Max.	Avg.	Max
Competing SO	Min	Avg.	Min	Max	Max.	Max.	Max.
Competing NSO	Avg.	Min	Avg.	Avg.	Avg.	Max.	Avg.
Goal Oriented	Avg.	Avg.	Max.	Avg.	Avg.	Min.	Avg.
Ad Hoc	Max.	Min.	Max.	Min.	Avg.	Min.	Min.
Domain Oriented	Avg.	NA	Avg.	Min.	NA	Avg.	NA

8 A Prototype System

For simulation purposes we have implemented a prototype using JXTA. We provide an option for creating Groups and Peers along with their properties, as shown in Figures 4a and 4b. This description is used to specify membership criteria. When a Peer applies for membership, its description is matched with the description of the group.

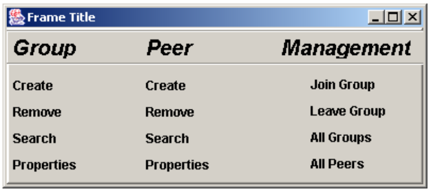
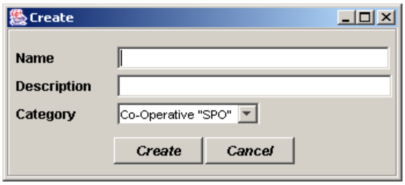


Fig. 4a. Main Menu



Name

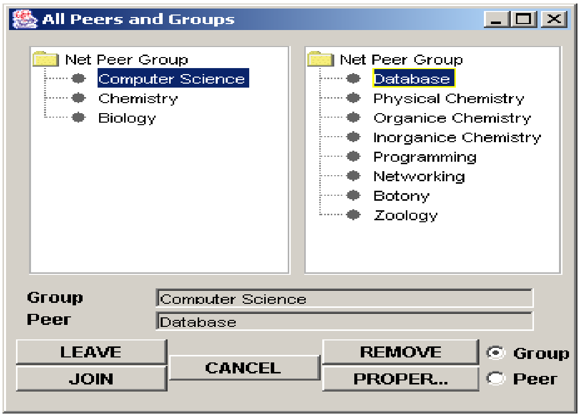
Description

Category Co-Operative "SPO"

Fig. 4b. GUI to create Group

Each JXTA Group is created with a randomly generated External Rating. A Group assigns randomly generated Internal Rating to all of its members at the time of membership. Each JXTA Group has a sorted list of its member Peers and each Peer has a sorted list of Groups to which it belongs. Peers apply for membership based on a high external rating of a JXTA Group. A JXTA Group grants membership based on overall rating of Peer (average of (IR of Peer * ER of Group)) and description of Peer. At any time a Group can have five members and any Peer can be a member of three different Groups. Peers can be added in different Groups using the interface but membership will be awarded based on the selection criteria of that specific Group.

Similarly, any Peer can resign from a Group at any time. Graphical User Interface for adding and removing Peer/s from Group/s is shown in Fig. 5a and member peers of Group Computer Science in Fig. 5b.


All Peers and Groups

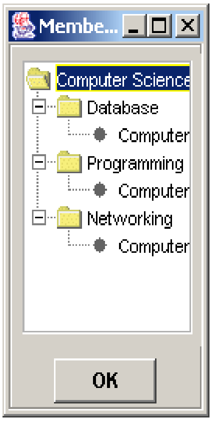
Net Peer Group
- Computer Science
- Chemistry
- Biology

Net Peer Group
- Database
- Physical Chemistry
- Organice Chemistry
- Inorganice Chemistry
- Programming
- Networking
- Botony
- Zoology

Group: Computer Science
Peer: Database

● Group
○ Peer

Fig. 5a. GUI for managing Group/s


Membe...

Computer Science
- Database
  - Computer
- Programming
  - Computer
- Networking
  - Computer

Fig. 5b. Members of Group

Each Peer has its own thread and after a certain time interval it discovers new Groups from the local cache of the JXTA environment and applies for membership. Peers will apply for membership to only that Group which has a high External Rating, as compared to the Group/s to which it already belongs. Peers keep on looking for the best Group and on discovering any suitable Group resign from the lowest rated Group. Similarly, each Group prefers to have highly rated Peers and on the membership of any new highly rated Peer cancels the membership of existing Peer with lowest rating. Result of this simulation was quite encouraging and as expected in the beginning the system has Groups and Peers attached without any uniform pattern but with the passage of time the system achieved stability and Groups with high rating have highly rated peers. Once the whole system is stable creating new Groups or Peers does not affect the overall membership of the Groups and Peers. Changing the description of either a Peer or a Group de-stabilizes the system, but as Groups have their own thread and constantly keep on comparing their description with member Peers – and in case of no match they cancel the membership of the Peer. This de-stabilization is temporary and the system tends to achieve its stable state in only a few iterations. Number of iterations to achieve stable state depends on the rating of a Group or a Peer, higher the rating of a Peer or a Group quicker the stabilization is achieved.

For each Group and Peer structured document is created which has description and rating of the peer. This structured document is used for creating credentials for the Peer, which is required to authenticate the Peer by a Group at the time of membership. During experimentation there are mainly three activities going on which are following:

1. Search of new Groups by Peers
2. Request for membership to those Groups
3. Membership acceptance by the Group

In the beginning all three activities are quite frequent as a new Peer applies for membership without any specific selection but with the passage of time Peer becomes more selective and apply for the membership to only those Groups which have higher rating than member Group with minimum rating. Similarly with the passage of time the Groups also become more selective in accepting the membership. As system becomes more stable there is even decline in the request for membership from Peers although this is true only for Peers with high ranking as they are already in highly ranked Groups but for Peers with low ranking the process of discovery of new Group always follow the membership request. It is noticed that Peers with high ranking end up in the Groups with higher ranking and Peers with low ranking end up in the Groups with low ranking, from which it is concluded that ranking of Group is indication of the member Peers ranking. In the simulation Peers are not caching any type of information about Groups with failed membership and thus keep on applying for the membership on each discovery of that Group but if Peers start caching the information of failed applications then there will be much more reduction in membership applications.

Each type of Group has it own membership policy which is implemented by corresponding java class extending net.jxta.membership.MembershipService. This java class has core membership methods i.e. apply(…), join(…) and resign(), which implements the membership policy of specific type of Group and update the member Peers for the corresponding Group. Membership policy of each type of Group can be easily changed by modifying the membership logic in apply(…) & join(…).

9 Conclusion and Summary

We present the concept of categorizing peers in communities on the basis of their expertise and interests. Social networks are a natural way for people to go about seeking information. Organizing peers in one form or another makes the discovery of resources efficient, whilst minimizing computational overheads. Categorizing the peers in communities is simple, open and easy to implement, and the initial overhead of developing communities pays-off latter at the time of resource discovery. Communities are more stable, and stability increases with the passage of time, have a simple learning time and are more adaptive to operate in a dynamic environment. We have proposed the external and internal rating for communities and peers respectively which may be used to support a given Quality of Service, effective participation of autonomous peers and better interaction among communities and member peers. Finally, we discuss the different services required to manage the group and requirements of the member peers. A JXTA implementation of a prototype system is discussed to describe the salient features of our approach. A key theme of this work is to determine how communities should be structured to support resource discovery, and how particular roles within a community can be used to determine interactions between participants within a community, and those between participants across community. This work extends techniques and results discussed in [14].

References

1. Karl Aberer: P-Grid: A Self-Organizing Access Structure for P2P Information Systems, Proceedings of the Sixth International Conference on Cooperative Information Systems (CoopIS 2001), 2001
2. S. R. H. Joseph, "Adaptive Routing in Distributed Decentralized Systems" http://www.neurogrid.net/publications/publications.html (2001)
3. Bin Yu, Mahadevan Venkatraman and Munindar P. Singh: 'An Adaptive Social Network for Information Access: Theoretical and Experimental Results', *Journal of the Applied Artificial Intelligence*, Volume 17, Number 1, (2003) 21–38
4. Leonard Foner. Yenta: 'A multi-agent, referral-based matchmaking system'. In *Proceedings of the 1st International Conference on Autonomous Agent*s, (1997) 301–307.
5. Bin Yu and Munindar P. Singh: 'Searching Social Networks', *Proceedings of Second International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2003, to appear.
6. Leonard Foner: *'Clustering and Information Sharing in an Ecology of Cooperating Agents, or How to Gossip without Spilling the Beans'*, Conference on Computers, Freedom and Privacy, 1995
7. Davis, R. and R. G. Smith: 'Negotiation as a Metaphor fro Distributed Problem Solving'. Artificial Intelligence (1983) 20, 63–109
8. Parunak, H. V. D.: 'Distributed Artificial Intelligence', Chapt. Manufacturing Experionce With the Contract Net, pp. 285–310, Research Notes in Artificial Intelligence. Los Altos, CA: Morgan Kaufmann Publishers, 1987
9. Christopher H. Brooks and Edmund H. Durfee: 'Congregation Formation in Multiagent Systems'. to apear in *The Journal of Autonomous Agents and Multiagent Systems*, early 2003
10. Pinar Yolum, Munindar P. Singh: '*Emergent Properties of Referral System'* Second International Joint Conference on Autonomous Agents and Multi-Agent Systems, 2003, to appear
11. Li Gong, "Project JXTA: A Technology Overview" http://www.jxta.org/project/www/docs/jxtaview_01nov02.pdf , October 2002
12. RFC 2768, "Middleware components" http://www.ietf.org/rfc/rfc2768.txt?number=2768
13. Mike Carew, "Anatomy of Components" http://www.middleware.net/components/articles/anatomy.html. ComponentFocus
14. Steven Lynden and Omer Rana, "Coordinated Learning to support Resource Management in Computational Grids" Second International Conference on Peer-to-Peer Computing, (2002) 81–89
15. FermiLab, "The DZero Project". See Web site at: http://www-d0.fnal.gov/