Reputation Measures based on Social Networks metrics for Multi Agent Systems[1]

Josep M. Pujol , Ramon Sangüesa

Software Department
Universitat Politècnica de Catalunya
Campus Nord, Mòdul C6, Despatx 204
C/Jordi Girona Salgado, 1-3
08034 Barcelona
{jmpujol, sanguesa}@lsi.upc.es

Abstract

The problem of calculating a degree of reputation to agents acting as assistants for the members of an eletronic community is presented. Usual reputation mechanisms rely on feedback after interaction between agents. An alternative way to establish reputation is related with the position of each member of the community within the corresponding social network. We propose a method based on this idea, discuss its properties as well as experimental results and compare them to other reputation mechanisms for electronic communities supported by agents.

Keywords: Social Networks, Electronic Communities, Multiagent Systems, Reputation, Trust, Reputation mechanisms.

1 Introduction

Electronic communities evolve around a given area of activity or topic of interest. The basis for their sustainability and persistence over time is the interchange of services by the community members. Members of electronic communities can be people or agents acting on their behalf. Typical interchanges include commercial transactions, documentary information and knowledge exchanges, responses to answers, advice, suggestions, help, reference to furhter information, etc. As well-known examples of each type of communities one can cite eBay, Firefly, or Knowledge Exchange We focus our discussion on communities created for knowledge management and organizational learning purposes, specifically those ones supported by a multiagent system where each agent represents a single user. Given that such communities survive through service interchange or mutual support, it is crucial for each member to be aware of the trust that could be given to other members as well as their reputation. These two characteristics are inherenttly dynamic and some schemas have been devised in order to keep track of them for several types of communities. We present in this paper an alternative approach based on the ubication of each user in the community social network. In section two, the characterisitics of multiagent systems for community support are presented together with currently used mechanisms for trust and reputation maintenance. A characterization of social networks is given in section three that also analyzes its propierties. Section four describes a new reputation mechanism for this type of multiagent systems. Section five presents a set of experiments on a concrete knowledge sharing community and Section six closes by discussing results and pointing to further work.

2 MAS for community support and their reputation mechanisms

Knowledge management in collaborative environments involve the interchange of information and knowledge among the members of a community. The knowledge management cycle involves detecting when new knolwedge is generated, who may be interested about it and delivering this knowledge to that people. Several approaches have been proposed in order to develop systems that detect user's knolwedge needs. Usually they are based on measuring the similarity in the competences of community members. This requires the maintenance of some type of user

[1] The research leading to these results have been supported by the Catalan Autonomous Government DURSI project i2CAT.

profile. Several multiagent systems have been developed to serve for these purposes. The agents act on behalf of community members, maintaining profiles or routing questions to other members' agents as is done, for example, in MARS [26] or the i2CAT Collaboratory [21,22]. Communities based on users preferences also share some treats with these systems. For example, agents in Firefly [9], Yenta [10] use profile similarity as a criterion for finding possible partners, which is at the basis for locating expertise within a community [23]. This same idea is used by recommender systems based on collaborative filtering [17]. In the case of knowledge sharing communities fromed by a set of people with expertise in a given domain not only it is important that agents be able to detect which people (represented by other agents) possess the adequate expertise for solving the problem at hand but it is also crucial to assess to which extent some of the members are recognised as experts by their colleagues in the community. The trust and reputation of experts has been typicallly assessed as a function of the quality of their response to knowledge requests coming from other members in the community. This is the schema used in some organizational learning systems as, for example, *Answer Garden* [1] and some commercial knowledge communities [4,7,8], see [11] for a discussion of the different ratings that can be obtained by analyzing response quality. All these systems rely on feedback (in the form of a rating) from the person receiving the response to a previous. Combining these ratings for each experts a numerical value for his reputation is calculated. The reputation measure gives an idea of the confidence one can have on an the quality of an expert's reponses and serves as a basis for guiding the search for experts. The disadvantage of this type of reputation mechanism is that it needs the explicit involvement of users in the form of ratings. This implies that a good reputation calculation and maintenance depends critically on the involvement of users and continued contribution of ratings. Normally insufficient users or low participation are at the root of bad collaborative filtering performance. Continous requierments for rating also create fatigue in users who tend to end up not issuing ratings at all, see a discussion about this in [12,18]. So, alternative less invasive methods in and less demanding in terms of their involvement are of interest This of course, raises the question of how can reputation be measured in the absence of any user feeback for expert's reponses. This goal is also important in devising reputation mechanisms for other types of agent systems. See [20] for a general discussion of trust and reputation in MAS [27] for a similar discussion in a more restricted type of community dealing with e-commerce . We put forth a system that takes into account the social network of the community.

2.1 Social networks

A *social network* [24] is a representation of the relationships existing within a community. Even within the same community several types of social network can be built depending on the social relationship taken into account: kinship, acquaintanceship, friendship, mutual support, cooperation, similarity are typical criteria used in establishing the social relationship components of a community. The corresponding social networks are represented as graphs. The construction of social networks for electronic communities helps in mapping the relationships among people that may not be aware of being related, given the special type of detached interaction peculiar to online communities. The basis for reconstructing this social interactions are the usage of the tools that give support to the online community which invole to some degree some public information about each member [6] . See *Referralweb* for a multiagent system supporting that type of community mapping [13] and MARS [26] or NetExpert [19] for multiagent systems supporting knowledge sharing communities on the basis of the corresponding social network strucuture.

2.2 Social Network structure and reputation

The ubication of a given member of a community within a social network can be used to infer some properties about his or her degree of expertise, i.e., his or her reputation. Experts who are well-known and highly regarded by most other members of the community tend to be easily identified as highly connected nodes in the global graph [2,4,15]. This relation infromation could be a basis for a reputation mechanism that used by users' assistant agents instead of having to resort to explicit ratings issued by each user. We describe in the following how this approach has been tested in the context of a multiagent system giving support to a knowledge sharing community.

3 Social Networks in a Collaboratory

The Collaboratory is a multiagent system supporting a research community. For a description detailed description, see [21,22]. The users of this system are currently the researchers involved in the Internet2 project in Catalonia, i2CAT *(http://www.i2.-cat.net).* They share a document repository and use a set of collaborative agents to obtain recommendations or for finding experts in a given topic within the community. Central to these functionalities is an agency that builds and maintains the social network of the community [16,19]. The social network building agency uses information located in he community members personal webpages. This information can be complemented with the knowledge contributed by users to the rest of the Collaboratory as well as the knowledge they extract from it. This is done by resorting to several user profiles. The details of their construction can be seen in [18]. For the purposes of the experiments discussed in this paper the test community has been extenced to all members of the Software Department of the Technical University of Catalonia.

Each kind of community may need its particular sources of information [24], for social network construction. Some of them are:

- Personal web pages
- Reports or documents authorship
- Participation in a project
- Hierarchical structure in the community or organization.
- Sharing of physical resources such as office, coffe boxes, etc.
- Sharing of virtual resources such as news group, irc channel, forum, etc.
- Email traffic, analyze emails is a ethical question that will not deal here, to walk arround we can use only the flow of emails not the content of them.

3.1 Building the Social Network

In order to build the social network for the present experiment it was necesary to create knowledge and content models of users' personal webpages by using *WebMining*, which is a agent that builds generic models from web pages, see [16,19]. For simplicity purposes one can see that model as a set of *terms* describing the knowledge contained in the personal pages. Once these models are obtained for each member of the community the Social Network is built by means of a similarity function which takes into account the strength of association between any two members of the community. Members of the community appear as nodes in the social network graph and the directed edges are calculated as follows.

$$w(a \to b) = w_{email}(a \to b) + w_{link}(a \to b) + w_{name}(a \to b) \quad (1)$$

The weight of a relationship is the sum of three factors:

$$w_{email}(a \to b) = \exists email(a \to b) \quad (2)$$

$email(a® b) is 1 when email address of member *b* exists in the web pages of member *a* , 0 otherwise.

$$w_{link}(a \to b) = \sum_{\forall a \in R(a,b)} \frac{2}{depth(a,a) + depth(a,b)} \quad (3)$$

R(a,b) is the set of resources (files than can be reached through the Web) that belong to member *b* and they have been found in the personal web page of member *a*.

depth(μ,x) is the depht of the resource *μ* in the personal web page of member *x*.

$$w_{name}(a \to b) = \frac{1}{2} \sum_{i=0}^{i \le \#name(a \to b)} \frac{1}{i^2} \quad (4)$$

#name(a® *b)* returns the number of occurrences of the name of member *b* within the personal web pages of member *a*. Extracting the name of a member of the community from the personal web page of another one is not trivial. Web pages are plain text information with html tags, there is no semantic information about names, thus a set of rules has to be used to disambiguaste names, see [16] for the deails.

A second, undirected graphs, results from this step also. Their weight are calculated as follows:

$$w(a \leftrightarrow b) = w(a \to b) + w(b \to a) \quad (5)$$

3.2 Social Network is a Small World

Figure 1 shows a partial view of the undirected Social Network for the UPC Software Department.

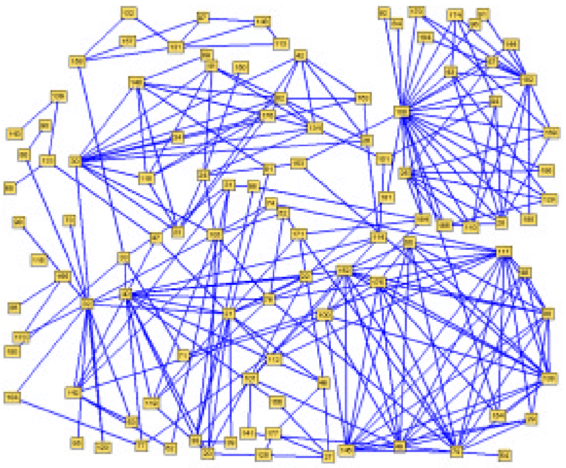

Fig 1: Fragment of Social Network

Table 1 shows the most important charasteristics of the undirected Social Network.

To formalize the notion of a small world, Watts and Strogatz [25] define the clustering coeficient C, and the characteristic path length L. We take the diameter of a graph as well D. [3] Clustering coeficient is a real in the inteval [0..1].

	Social Network	Random Network
#nodes	139	139
#edges	394	394
C	0.7045	0.3774
L	3.6572	3.016
D	9	6

Table 1: Social Networks characteristics

Watts and Strogatz define a small world graph as one is which L³L_{rand}, or L»L_{rand}, and $C>>C_{rand}$ where L_{rand} and C_{rand} are the characteristic path length and clustering coeficient of a random graph with the same number of nodes and edges. Using Watts and Strogatz definition Social Network can be considered as a small world.

4 Using Social Network metrics for reputation measurement

NodeRanking is our proposal for creating a ranking of reputation ratings of community members by means of the corresponding Social Network.

The rating that *NodeRanking* creates is based on the idea that each node on the graph has an associated *degree of authority* that can be seen as an importance measure. Initially, all nodes are assumed to have the same authority. After running *NodeRanking* the resulting authority measure is used to infer the reputation of a node within the graph, that is, the reputation of a member within his community.

Authority of a node, *a*, is calculated as a function of the total measure of authority present in the network and the authority of the nodes pointing to *a*. If a node is not referred by any other node in the network it is assigned a default authority value. Authority values are a positive values.

The algorithm for authority calculation is inspired in the ranking algorithms for web pages based on web topology [14,15]. The idea is to apply a similar reasoning about link topology in webpages to a the topology of a Social Network, i.e., the link topology of a directed graph. In a directed graph the edges have a direction, the out-edges of a node are the edges that start in this node, the out-nodes are the nodes that can be reached through out-edges.

The main idea of the *NodeRanking* algorithm is that each node has an authority and a part of its authority is propagated to the out-nodes via out-edges.

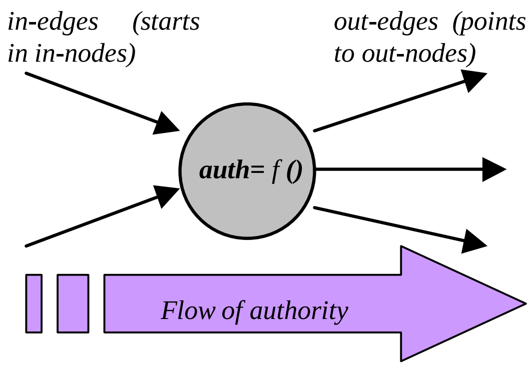

Fig 2: Flow of authority in *NodeRanking*

The authority of a node depends on the authority of its in-nodes. The authority of one of the out-nodes in figure 2 node depends in part on the authority of this node. Cycles in the graph can produce critical deadlocks in the calculations. The *NodeRanking* algorithm overcomes these problems and insures the convergence of calculations.

4.1 The *NodeRanking* Algorithm

NodeRanking follows essentially the *random walker strategy* to explore the graph,. It starts in a randomly selected node, and proceeds by selecting one of the nodes that can be reached through out-edges.

```
do
   v ¬ randomSortNodes()
   do
      n ¬ getNode(v)
      do
         nnew ¬ getNextNode(n)
         if ($nnew) ® passAuthority(nnew,n)
         n ¬ nnew
      while ($nnew)
   while (∅ empty(v))
while (∅ converge( ))
```

Fig 3: *NodeRanking* algorithm

We have used some functions that require additional explanation.

randomSortNodes ( ): returns a vector of all nodes of the graph ordered randomly without repetitions. In each iteration the algorithm will visit all the nodes of the graph without an established order.

getNextNode(node x): returns one of the out-neighbours nodes of the node *x*. Each node has a set, that can be empty, of out-edges that points to other nodes. This function can return a null node to stop the path of the random walker by introducing some elements of randomness. When *getNextNode()* returns a null node, in fact, the path is broken. There are two cases where the path is broken: 1) If the algorithm arrives at a node that has been already visited in the previous *k* steps. The algorithm keeps a window of *k* elements and breaks the path if a new node already exists in the window. 2) The walker evaluates the jumping probability whenever it reaches a node. This probability is a function of the connectivity of the node. The walker skips the node with a probability $\Pr_{jump}(a)$.

$$\Pr_{jump}(a) = \frac{1}{2(\#out-edges(a))^2} \qquad (6)$$

Nodes with fewer out-edges have a greater probability of breaking the path. This could be seen as a walker that gets bored because of the little range of choices. When *get*N*extNode*( ) returns the next node *b* to be visited from *a*. This node is selected with a probability that is calculated as a function of the weight of the edge between *a* and *b*, the corresponding density probability function is shown below.

$$\Pr_{choose}(a \to b) = \frac{w(a \to b)}{\sum_{\forall a \in out-nodes(a)} w(a \to a\,)} \qquad (7)$$

Where w (a -> b) is the weight of the link connecting a and b,.

passAuthority(node x, node y): this function assigns part of the authority of node *y* to node *x*. The next equation shows the change of authority.

$$auth(y) = auth(y) + \left[\frac{\Pr_{choose}(x \to y)auth(x)}{F_y}\right] \qquad (8)$$

Where *auth(y)* is the authority of the node *y* and F_y is a factor to maintain the value of authority within a limited range of values. Without this factor, values calculated with Equation 8 would tend to infinity because the authority of a node gets bigger and bigger as the method proceeds. The factor *F* controls the growth of the authority and also reduces the effects of randomness. Each node has its own *F* factor that remembers the state of the total authority value of the graph the last time that the node was involved in a *passAuthority* function call. These factors depend on the authority value of all the nodes in the graph at the time that the authority of node *y* was updated. The factors of more frequently visited nodes grow faster than the values of the less visited ones. The growth of these factor's values is monotonously increaing. Thus we can insure convergence towards a finite value. Without this factor nodes that have been last in the random selection would have an advantatge over the other ones, because the graph is accumulating more and more authority as the algorithm proceeds. Factor F is initialized for every node as the sum of the authority of all nodes in the graph. The initial authority of a node has to be positive, and factor F has to be bigger or equal than 1.

converge( ): this function is evaluated each time that the algorithm has visited at least one time every node in the graph. This function creates a ranking of the nodes by authority, and then compares it with the last p rankings obtained in previous iterations. To compare them the correlation coefficient is used.

$$converge \leftarrow \left(1 - \frac{1}{p}\sum_{i=0}^{i<p} correl(r_i, r_{i+1})\right) \leq j \quad (9)$$

Where, j is a threshold value. Figure 4 shows the convergence of the algorithm for different graphs. G_1 has 139 nodes and 467 edges, and G_2 has 139 nodes and 311 edges. The parameters of the algorithm for this figure are fixed at k=4, p=30, $auth_{initial}$ = 1 and j=0.0001.

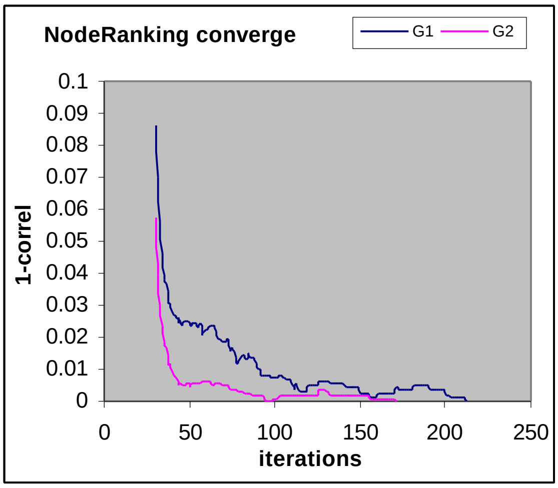

Fig 4: *NodeRanking* algorithm convergence.

NodeRanking does 212 iterations for G1 and 171 iterations for G2.

5 Experiments about ranking by reputation

The *NodeRanking* algorithm was applied to the Social Network of the experimental community formed by the members of the Software Department. We present in table 2 the rankings obtained for 34 randomly selected members of the community with their names omitted. In order to validate this ranking as a correct reputation measure they were compared against a real and well-agreed measure of reputation. This community being a research community that produces scientific publications a good way to establish this comparison is to resort to the values calculated for those 34 members by an independent ranking agency for scientific publications. That is , the citation indexes for each of the 34 randomly selected members of the Software Department were compared against the rating that *NodeRaking* yielded, $Rank_{NodeRanking}$. Citation indexes are a clear indicator of reputation in scientific communities. *CiteSeer* [5] was used as a source for citation index values. In *CiteSeer* the papers of a researcher can be retrieved and as well as the corresponding number of citations and self-citations for each of his papers. We have built two rankings of the chosen members of the community. $Rank_{cite}$, reflects sorts researchers by the number of citations and $Rank_{cite\text{-}self}$, sorts researchers by number of citations without counting self-citations.

Exp. ID	R_{cite}	$R_{cite\text{-}self}$	$R_{NodeRanking}$	Exp. ID	R_{cite}	$R_{cite\text{-}self}$	$R_{NodeRanking}$
60	1	1	4	23	25	26	10
31	2	2	16	46	30	29	27
131	3	3	2	38	31	31	22
41	4	4	7	190	32	32	32
28	6	5	5	98	19	20	15
32	7	7	1	146	5	6	11
112	11	8	25	189	18	18	18
21	8	10	3	174	16	15	14
27	9	11	12	97	27	27	20
129	12	9	13	25	15	12	24
33	13	13	6	180	28	28	28
22	14	16	8	104	20	19	29
44	10	17	21	188	33	33	33
182	23	22	26	133	29	30	34
20	21	23	9	164	34	34	31
42	24	24	23	173	22	21	19
165	26	25	30	176	17	14	17

Table 2: Table of results.

In order to compare the quality of the rating obtained by the *NodeRanking* algorithm two similarity measurements were used: 1) Correlation coefficient between $Rank_{NodeRanking}$ values and $Rank_{cite}$, and $Rank_{cite\text{-}self}$, 2) Distance between permutations of those rankings. The permutation distance is very similar to euclidean distance, the rankings are represented by vectors.

$$d_{perm}(\vec{R},\vec{P}) = \sum_{i=1}^{i\leq card(\vec{R})} (R_i - P_i)^2 \quad (10)$$

Table 3 shows the correlation values between $Rank_{cite}$, $Rank_{cite\text{-}self}$ and $Rank_{NodeRanking}$.

Correl($Rank_{cite}$,$Rank_{cite\text{-}self}$)	0.983
Correl($Rank_{cite}$,$Rank_{NodeRanking}$)	0.773
Correl($Rank_{cite\text{-}self}$,$Rank_{NodeRanking}$)	0.741

Table 3: Correlation values between rankings

Figure 5 shows the correlation between $Rank_{cite}$ and $Rank_{NodeRanking}$.

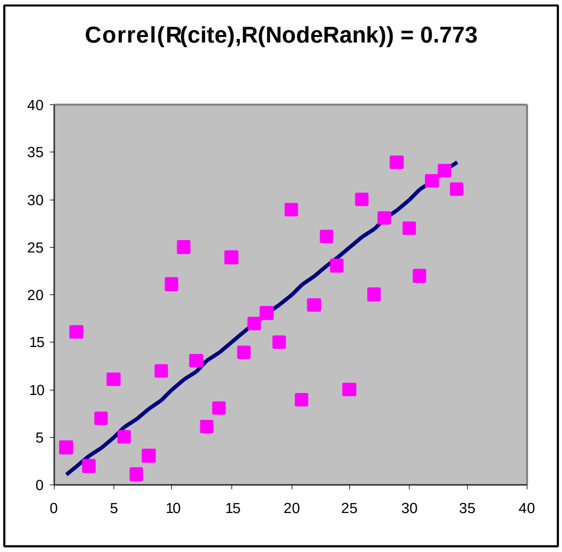

Fig 5: Correl($Rank_{cite}$,$Rank_{NodeRanking}$)

Figure 6 shows the correlation between $Rank_{cite\text{-}self}$ and $Rank_{NodeRanking}$.

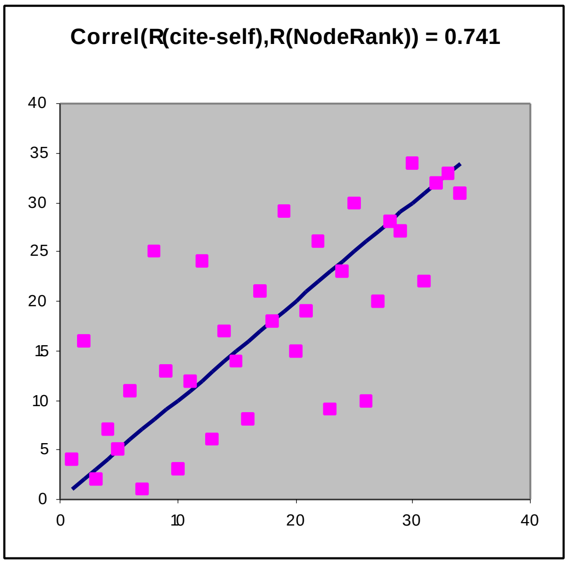

Fig 6: Correl($Rank_{cite\text{-}self}$,$Rank_{NodeRanking}$)

As it can be seen, the correlation values are 0.74 and 0.77 comparing against total number of citations or the number of citations without self-citations. This seems to indicate that both measures a quite strongly correlated.

As a ranking is not strictly a distribution it is worth having a look at the results for the permutation distance measure. We have introduced a normalization factor calculated for the case of maximum disorder in the rankings. The maximally disordered ranking, gets a value of 1. A perfect ranking would get a 0 value. A random ranking would obtain a value of 0.5. Table 4 shows the values of the normalized permutation distance.

d_{perm}($Rank_{cite}$,$Rank_{cite\text{-}self}$)	0.009
d_{perm}($Rank_{cite}$,$Rank_{NodeRanking}$)	0.1134
d_{perm}($Rank_{cite\text{-}self}$,$Rank_{NodeRanking}$)	0.1295
d_{perm}($Rank_{cite}$,$Rank_{Random}$)	0.5

Table 4: Permutation distances between rankings

The results of permutation distance measure are equivalent to the results of correlation coeficient.

6 Discussion and further research

The values of the ratings calculated by the topological information of a community's social network has been used as a measure of each member's reputation. A new algorithm *NodeRanking* has been devised to obtain such measures. An experiment has been made on a real community and the results compared against a well-agreed valid measure of reputation for that type of communities. The results seem to indicate that *NodeRanking* values are a good approximation to reputation measures.

The proposed criterion for reputation measurement in a collaborative multiagent system and the corresponding method have as an advantage with respect to other ones [27] the fact that it does not require to have users continously and explicitly issuing ratings, a method that is seen as a burden on users and eventually a reason for poor performance of collaborative systems.

Another advantage of *NodeRanking* is that the proposed method is solely based on topological infromation, thus making complete abstraction of any other information. On the other hand its success hinges critically on the quality of fit between the social network representation and the real community structure. In the case of the Social Network of the Collaboratory it seems that the information used to build the social network is well-suited to research communities.

Other multiagent system methods that use social networks either don't use them for reputation measurement, as is the case of *ReferralWeb* [13] or still rely exclusively on rating feedback from users as *MARS* does [26]. This last one (as [27] does) has only been tested on a simulated community as opposed to a the test we carried on a real one.

Further experimentation will be carried on with other types of knowledge sharing communities in order to test the dependence between the information used in building the social network and the final quality of the reputation measurements obtained from it.

7. References

[1] M. S. Ackerman and D. W. McDonald. Answer garden 2: Merging organizational memory with collaborative help. In *Computer Supported Cooperative Work*, pages 97–105, 1996.

[2] L. Adamic and E. Adar. *Friends and neighbors on the web*, 2000.

[3] L. Adamic. The small world web. In S. Abiteboul and A.-M. Vercoustre, editors, *Proc. 3rd European Conf. Research and Advanced Technology forDigital Libraries, ECD*L, number 1696. Springer-Verlag, 1999.

[4] AskMe. Available on http://www.askme.com/.

[5] CiteSeer. Available on http://www.citeseer.com/.

[6] N. S. Contractor, B. J. O'Keefe, and P. M. Jones. *IKNOW: Inquiring knowledge networks on the web.* Computer Software Deparment, University of Illinois, 1997.

[7] Experts Exchange. Available on http://www.experts-exchange.com/.

[8] Exp.com. Available on http://www.exp.com/.

[9] Firefly. Available on http://www.firefly.net/.

[10] L. Foner. *Yenta: A multi-agent, referral-based matchmaking system*, 1997.

[11] R. Garcia. *Extensió col.laborativa del servei de localització d'expertesa.* Master's thesis, Technical University of Catalonia, 2001.

[12] Grasso, A. Mixing Cognitive and Collaborative Filtering. *Proceedings of the I3net Community of the Future Conference*. Sienna, Italy, October 1999.

[13] H. Kautz, B. Selman, and M. Shah. The hidden web. *AI Magazin*e, (18), 1997.

[14] J. Kleinberg. *Authoritative sources in a hyperlinked environment.* Technical Report RJ 10076, IBM, May 1997.

[15] L. Page, S. Brin, R. Motwani, T. Winograd, *The PageRank citation ranking: Bringing order to the Web,* submitted for publication.

[16] J. M. Pujol. *Netexpert: Localitzador d'expertesa.*Master's thesis, Technical University of Catalonia, 2000.

[17] R. Sangüesa, A. Vázquez, and J. Vázquez. Ace: A multiagent recommender system using
mixed collaborative and cognitive filtering. In *Proceedings of the WARS 2000, Workshop on Agent-Based Recommender System*s, 2000.

[18] R. Sangüesa, A. Vázquez and J. Vázquez. *The ACE Recommender System.* Technical Report LSI-01-21-R., Technical University of Catalonia, 2001.

[19] R. Sangüesa and J. M. Pujol. NetExpert: A multiagent system for expertise location. *Accepted to IJCAI'01 Workshop on Organizational Memories and Knowledge Management*, 2001.

[20] M. Schillo, P. Funk, M. Rovatsos. Who can you Trust: Dealing with Deception. In C.

Castelfranchi, Y. Tan, R. Falcone and B. S. Firozabadi (ed.), *Proceedings of the Workshop "Deception, Fraud and Trust in Agent Societies" of the Autonomous Agents Conference*, 1999.

[21] A. Vázquez, J. Vázquez, I. Barrio, J. M. Pujol, and R. Sangüesa. *The collaboratory*. Technical Report LSI-01-22-R., Technical University of Catalonia, 2001.

[22] A. Vázquez, I. Barrio, J. Vázquez-Salceda, J.M. Pujol, R. Sangüesa. *An agent-based Collaboratory*. Accepted for publication at ACAI'01, Advanced Course on Artificial Intelligence, Prague, July 2001.

[23] A. S. Vivacqua. Agents for expertise location. In Proceedings of the 1999 AAAI Spring Symposium on Intelligent Agents in Cyberspace, 1999.

[24] S. Wasserman and J. Glaskiewics. *Advances in Social Networks Analysi*s. Sage Publications, 1994

[25] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Natur*e, (393), 1998.

[26] Bin Yu and Munindar P. Singh, A Social Mechanism of Reputation Management in Electronic Communities. *Proceedings of Fourth International Workshop on Cooperative Information Agents*, pages 154-165, 2000.

[27] G. Zacharias and P. Maes, Trust Management Through Reputation Mechanisms, *Applied Artificial Intelligence* 14, pp 881-907, 2000.