Application of Agent-based Personal Web of Trust to Local Document Ranking

Marek Kopel and Przemysław Kazienko

Wrocław University of Technology, Institute of Applied Informatics
Wyb.Wyspiańskiego 27, 50-370 Wrocław, Poland
{marek.kopel, kazienko}@pwr.wroc.pl

Abstract. Web is the boundless source of information and no one is able to process the vast amount of new documents published on the web every day, even with filtering out the documents the user is not interested in. However, most of the recent web documents are blog posts, news and other documents with the author information established. Each author who is also the receiver of web documents possesses their own personal agent that delivers trust information related to other authors as well as rank data for each new document. Trusts and ranks available for agents are exchanged between them and in this way new authors and new web documents can be easily assessed. Based on the general concept of Web of Trust the new idea of Personal Web of Trust and its application to local ranking method for web documents is proposed in the paper.

1 Introduction

The blogosphere's rapid expansion, which enables an easy way of publishing documents on the web, caused a great growth of people who became authors of web documents. The large amounts of new authors create new documents in different styles and with varying credibility for a reader. The reader, i.e. web user, trying to reach valuable and reliable information, needs to make some trust assertions concerning web documents and their authors. In order to help users to keep their trusts assertions up to date a concept of a Web of Trust (WoT) has been developed. Since blogs facilitate the mass communication and each user may be a blog author, many of the authors may be in same kind of relationship. Moreover, their relationship may influence their trust to others. The main idea of WoT is to make use of users' trusts to known authors for inferring the trust to new, unknown users in order to estimate their trustworthiness and propose rank values of their documents for the reader.

Web documents may be enriched with semantic information about their authors' relationships. Among the standards allowing turning a web document into the Semantic Web document by adding the information on author relationship are FOAF [2] and XNF [1]. FOAF (Friend of a Friend) is a method based on linking from web documents to a machine-readable RDF file which describes its author and the people the author knows. By processing those RDF files an author relation network can be generated automatically. XFN (XHTML Friends Network), in turn, is a method for denoting the type of relationship between linking and linked documents' authors. The relationship description is an attribute of a XHTML link from one document to another but it regards the document authors. Since determining the authorship of a web document is not always an easy task, it is proposed to use the hybrid relationship description using the combination of FOAF and XFN.

The main application of the WoT presented in this paper is the document ranking and filtering in the collaborative way and its collaborative paradigm is based on the trust inferring. However, the idea of the Web of Trust presented in this paper shall be distinguished from the general idea of Web of Trust [19] used for identification and authentication usually based on PKI (public key infrastructure) [3, 6, 11]. The presented method uses its agent based WoT for achieving another goal but some ideas of binding public keys to users from PKI WoT may be also considered to expand the method functionality. Some research in the direction of combining FOAF semantics and PKI WoT has already been made and resulted in creation of Web of Trust RDF Ontology [4, 5]. In another approach authors considered joints of different trusts delivered by other users in the network, however distribution matters have not been studied [18]. Pujol *et al.* tried to extract information about trust to others from metrics within the social network created upon mutual communication [16]. This is approach similar to the analysis of user social position [10]. Kollingbaum and Norman investigated trust extraction from the business contract data and the negotiation process performed by contracting agents [12]. Guha *et al.* studied the problem of transitive trust and distrust and their propagation especially by means of different rounding approaches [7].

2 Trust to Authors

The concept developed in the paper covers all internet services in which there is a set of documents created by valid authors like personal web pages, blogs, emails, pictures galleries etc. Each document's author, who is simultaneously a system user, possesses one personal agent which is able to communicate with other user agents and performs several tasks for its owner (user). In this way we obtain a distributed multi-agent architecture. Each user of the system can create their own documents, e.g. posts on blog and publish them in the open environment. Other users may access all published documents but due to the possibly large number of new or updated items it is difficult for them to select only reliable documents for reading. The possible solution is to infer the document trust from the trust to the document's author. In this way, we can order the obtained documents according to the trust worthiness: the most trusted items are moved to the top of the list.

Each user agent is responsible for maintaining the set of trusts to other users, i.e. confidences in other users authoring. Additionally, the personal agent also ranks and orders all documents that have been delivered to the user. Both, trust to other users and ranks of documents, can be either manually delivered by the user – agent's owner or provided by the agent itself. The agent creates its trusts and ranks based upon the knowledge supplied by other agents. In this paper, by using terms *author*, *user* and *agent* we address the same technical concept. Even though the semantics of these terms may have different intuition, for the purpose of our research we assume that each author is a user, each user is a potential document author and each user has an agent which maintains the knowledge of its user trusts, document ranks and document authorships. That is why the three terms: author, user and agent are used here equivalently.

2.1 User Trust

Each human in the real world may have some relationships with other humans. They may be family, friends, colleagues or acquaintance. The quality of such relationship may express also the level of the trust that one man have to another. Note that the human relationships are usually asymmetric. Additionally, the trust can be derived from the personal assessment of documents authored by the certain user. We can have more or less confidence in the user who created documents we like. Hence, a user that is not in any relationship with any other user in the real world may trust in authors whose documents one found interesting. Based on the relationships or document evaluation people have available from the real world, they are able to manually estimate the value of the trust function to some another system users.

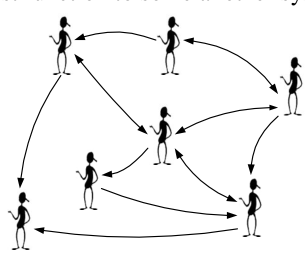

Fig. 1. Web of Trust. Each node represents an author and edges are trusts to author that can be derived from the real world relationships or from author's document evaluations.

Definition 1. User trust $T^{usr}(a_i \rightarrow a_j) \in [0,1]$ from author (agent) a_i to author (agent) a_j is the trust manually set or approved by author a_i. User trust $T^{usr}(a_i \rightarrow a_j)$ is retained by agent a_i in its internal database.

When a user sets the trust to another one manually, the latter is being included into the former's Personal Web of Trust (PWoT).

2.2 Personal Web of Trust

The general idea for Web of Trust is that trust may be interpreted as a directed edge incident with two nodes – authors, shown in Fig. 1. Direction of the edge denotes who trusts in who, i.e. an edge from a_1 to a_2 stands for "a_1 trusts to a_2 ".Now let us filter out the both direction edges and take only the outgoing edges from one node only. The edges that go out from user a_1 and the nodes incident with them form a directed graph representing the Personal Web of Trust of user a_1 who occupies the center-placed node, as shown in Fig. 2.

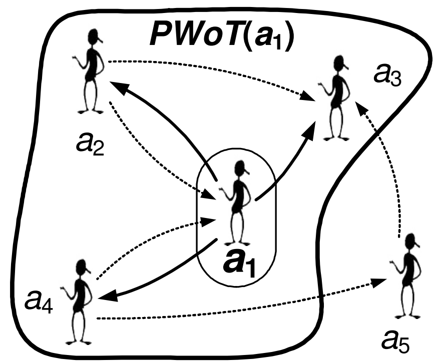

Fig. 2. Personal Web of Trust of author a_1 $PWoT(a_1)$ is a fragment of the entire Web of Trust surrounded by the solid line that includes only the authors whose trust to was assigned explicit by author a_1. Elements drawn with a dotted line does not belong to $PWoT(a_1)$.

Definition 2. Personal Web of Trust $PWoT(a_i)$ for author a_i is the set of other authors (agents) $a_j \in A$ from the entire author set A for which user trust $T^{usr}(a_i \rightarrow a_j)$ from a_i to a_j is known for a_i.

In other words, Personal Web of Trust of a single user a_i includes all the authors whose the user's trust to was assigned explicit by the given user a_i.

2.3 Agent Trust

Whenever an author a_i needs an information about another author a_j from the outside of $PWoT(a_i)$, the a_i's agent can propose the trust to author a_j based upon trusts delivered by other authors from $PWoT(a_i)$. To calculate the appropriate trust level for the new, unknown author a_j the agent a_i needs to exchange knowledge related to a_j with other agents from the $PWoT(a_i)$. Next, the agent trust $T^{agn}(a_i \rightarrow a_j)$ is evaluated and suggested to the user a_i.

Definition 3. Agent trust $T^{agn}(a_i \rightarrow a_j) \in [0,1]$ from author (agent) a_i to author (agent) a_j is the trust calculated by agent a_i, based on the trusts delivered by other agents from $PWoT(a_i)$, in the following way:

$$T^{agn}(a_i \rightarrow a_j) = \frac{\sum_{a_k \in PWoT(a_i)} T^{usr}(a_i \rightarrow a_k) \cdot T^{rsp}(a_k \rightarrow a_j)}{\sum_{a_k \in PWoT(a_i)} T^{usr}(a_i \rightarrow a_k)} \qquad (1)$$

where: $T^{rsp}(a_k \rightarrow a_j)$ is *response trust* to a_j provided by agent a_k:

$$T^{rsp}(a_k \rightarrow a_j) = \begin{cases} T^{usr}(a_k \rightarrow a_j), & \text{if } T^{usr}(a_k \rightarrow a_j) \text{ is known} \\ \lambda_k \cdot T^{agn}(a_k \rightarrow a_j), & \text{if } T^{usr}(a_k \rightarrow a_j) \text{ is unknown} \\ & \text{and } T^{agn}(a_k \rightarrow a_j) \text{ is known} \\ 0, & \text{otherwise} \end{cases} \qquad (2)$$

λ_k - the importance of the agent trust in relation to user trust for agent a_k; $\lambda_k \in [0,1]$.

The value of coeficient λ_k can be manually assigned by user a_k. The value λ_k=0.5 means that for user a_k others' trusts are half as important as a_k's own are.

To calculate the response trust $T^{rsp}(a_k \rightarrow a_j)$, the requested agent a_k may need to evaluate its own agent trust $T^{agn}(a_k \rightarrow a_j)$. For that reason the evaluation may become recursive. This causes a problem of generating a communication chain reaction to which the concept of time-to-live (TTL) may be a cure, discussed in the section related to agent communication.

2.4 Document Rank and Authorship

The main reason for maintaining *PWoT* is the ability to estimate a new document's rank. Rank is the reflection of user's opinion on the document in the aspect of relevance to user's interest. The document rank application allows user to access to the most relevant documents first, as estimated by their agent.

Definition 4. User rank $R^{usr}(a_i \rightarrow d_j) \in [0,1]$ from author (agent) a_i to document d_j is the rank manually set or approved by author a_i. User rank $R^{usr}(a_i \rightarrow d_j)$ is retained by agent a_i in its internal database.

User a_i does not rank documents which a_i is the author of, e.g. documents d_1 to d_4 created by user a_1 in Fig. 4. In this case we assume that the user rank is the highest possible. In agent knowledge the fact of being an author of a document is called authorship (Fig. 3). On the other hand, agent a_i retains also user ranks to documents created by others, e.g. agent a_1 preserves ranks to documents d_5 to d_8, Fig. 4c.

Definition 5. Author a_i's authorship of the document d_j denoted as $Ath(a_i \rightarrow d_j)$ is true only if a_i is the author of the document d_j.

Assignment of the rank to a document works the same way as a user trust assignment does. It can be done manually (explicit) by the user and then it immediately becomes the user rank. However, in case the user did not have a chance to see the document and rank it personally, the agent is responsible for delivering its agent rank, as the initial, suggested value.

Definition 6. Agent rank $R^{agn}(a_i \rightarrow d_j) \in [0,1]$ from author (agent) a_i to document d_j is the rank calculated by agent a_i, based on the ranks delivered by trusted agents from the $PWoT(a_i)$, in the following way:

$$R^{agn}(a_i \rightarrow d_j) = \frac{\sum_{a_k \in PWoT(a_i)} T^{usr}(a_i \rightarrow a_k) \cdot R^{rsp}(a_k \rightarrow d_j)}{\sum_{a_k \in PWoT(a_i)} T^{usr}(a_i \rightarrow a_k)} \qquad (3)$$

where: $R^{rsp}(a_k \rightarrow d_j)$ is *response rank* of document d_j provided by agent a_k:

$$R^{rsp}(a_k \rightarrow d_j) = \begin{cases} 1, & \text{if } Ath(a_k \rightarrow d_j) \text{ is true} \\ R^{usr}(a_k \rightarrow d_j), & \text{if } R^{usr}(a_k \rightarrow d_j) \text{ is known} \\ \tau_k \cdot R^{agn}(a_k \rightarrow d_j), & \text{if } R^{usr}(a_k \rightarrow d_j) \text{ is unknown} \\ & \text{and } R^{agn}(a_k \rightarrow d_j) \text{ is known} \\ 0, & \text{otherwise} \end{cases} \qquad (4)$$

τ_k – the importance of the agent rank in relation to the user rank for agent a_k; $\tau_k \in [0,1]$.

The value of τ_k is manually assigned by user a_k. The value τ_k=0.333 means that for user a_k others' document ranks are one third as important as a_k's own are.

Similarly to agent trust calculation, the agent rank estimation is also recursive. The same problem of chain reaction emerges. It shall be noticed that the evaluation of both agent trust T^{agn} and agent rank R^{agn} using one, common query may improve the performance.

3 Local Document Ranking

Building a Personal Web of Trust and providing user agents with the knowledge of author trusts and document ranks delivers each user an useful collaborative filtering tool. This tool may be applied to a collection of documents in many ways. One of them is the local ranking of documents while a user navigates across in the web. "Local" means created and proposed by user's own agent. Ranking is a collection of documents ordered according to document ranks and their author trusts.

The method presented below assumes the existence of a prior document list (or at least a document set) known by the user agent. This input list may be for example a response to the search engine query or an RSS file.

As stated above, each document d_j may have two values: $R^{usr}(a_i \rightarrow d_j)$ and $R^{agn}(a_i \rightarrow d_j)$ assigned in the agent a_i's database. The most evident way of making use of them is to sort input documents by their values to obtain the most and the less relevant documents. Obviously, the collection should exclude all documents the user created themselves since they are the most relevant. First, the user rank value $R^{usr}(a_i \rightarrow d_j)$ is taken into account at sorting since it reflects user interest the best. However, many documents, especially the new ones, will have $R^{usr}(a_i \rightarrow d_j)$ *unknown*. For such documents d_j the agent rank $R^{usr}(a_i \rightarrow d_j)$ is evaluated and utilized for ordering.

Two other values $T^{usr}(a_i \rightarrow a_k)$ and $T^{agn}(a_i \rightarrow a_k)$ may also be used for document d_j authored by a_k. They can be utilized at estimation of the d_j's position in the output document list. It appears to be more user friendly to present the author ranking first and next, after choosing an author, to show the ranked list of documents. Alternatively, one ranking may be proposed to the user with the several document lists sorted by each of the four values. For example, sorting the document collection descending by T^{usr} and then R^{usr} gives the ranking of most relevant documents from the authors with which the current user has the best relationship.

Yet another idea for personalizing of the document list is to sort them by a combination the four values. Moreover, the custom weighted average may deliver the best results.

4 Agent Communication

The agent approach to the Web of Trust assumes that the knowledge about the trust is distributed among all user personal agents and therefore agent communication for knowledge exchange is needed. This regards especially the information about trusts and ranks delivered or approved by users and stored in their agents' internal database. Additionally, each agent needs the knowledge about its owner's document authorships to be able to respond to rank queries and properly order the input collection of documents (see Fig. 3).

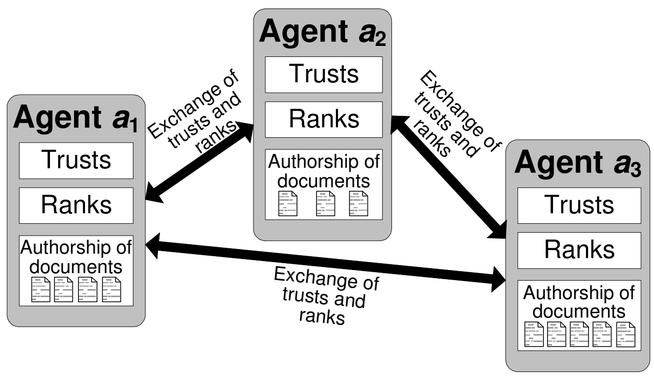

Fig. 3. Communication to the Web of Trust: agents exchange user trusts and document ranks and retain information about their owner's authorships of documents

The main problem in exchanging the knowledge by query is the, mentioned earlier, chain reaction issue. When an agent a_i evaluates its $T^{agn}(a_i \rightarrow a_j)$ or $R^{agn}(a_i \rightarrow d_j)$ for an unknown agent a_j or document d_j, it broadcasts a query to all other agents a_k from its $PWoT(a_i)$. Recursively, each of the asked agents a_k may not possess the appropriate knowledge, i.e. $T^{usr}(a_k \rightarrow a_j)$ or $R^{usr}(a_k \rightarrow d_j)$ are not available. In such case, agent a_k needs to evaluate its $T^{agn}(a_k \rightarrow a_j)$ or $R^{agn}(a_k \rightarrow d_j)$ and it also sends a new query to agents from its $PWoT(a_k)$. To prevent the system from circulating queries going around the web infinitely, each query should contain a time-to-live status (TTL). TTL is the number initiated by agent a_i that sends the primary query. Once agent a_k is queried for a trust or rank, it needs to forward the query among its $PWoT(a_k)$. It also decreases TTL with 1 for the relayed query. If agent a_k has received a query with TTL=1, then such query must not be forwarded any more and a_k responds with the value of 0. The initial TTL value is a system parameter common for all agents. Theoretically, six is the value that allows reaching the opinions of authors from the entire world – see the six degrees of separation hypothesis [13]. However, it is most likely that the only appropriate value that respects the performance limitations can be established empirically. It appears it should be about 2-3.

Another improvement of the performance is the blockade of the backward querying. For that reason, each query should have the source information attached – name of the agent who cast the original query. Just like TTL, this data should be forwarded in the recursive queries. When a query is received by an agent, it checks whether it has already received the same query. If yes, the query is dropped: not forwarded and not responded to.

There is also a serious problem related to delays in communication. How long an agent should wait for the query response (T^{rsp} or R^{rsp}) and how to process the responses? To address this problem we shall assume that the TTL of the forwarded query is always returned in a response. This allows estimation how far is an agent that has the user trust or rank available, i.e. how many agents on the return path influenced the response. An agent that queried its *PWoT* should only take into account the responses with the highest TTL. This assures the least influence of the opinions by authors which the user has not trusted directly. The response TTL value solves also another problem regarding the number of responses that need to be obtained by agent a_k before a_k evaluates the response and sends it back. Each agent should wait with the evaluation of its response until it receives a response with TTL lower than the very first response TTL. According to the intuition if we assume the architecture is in some degree homogenous the responses with higher TTL shall arrive earlier.

Even though the introduction of TTL is helpful, it is not enough. It is expected that some amount of the queries may be timed out because of the hardware limitations. To solve this problem the cache expiration period should be studied.

5 Scenario Example

Let us consider the initial Personal Web of Trust $PWoT(a_1)$ from Fig. 4a. The user a_1 goes online to find information on the favorite music band, types "*Pink Floyd*" into search engine and obtains a collection of relevant documents. Some of these documents are album and concert reviews. It turns out that many of the reviews (d_5, d_6) are blog posts by author a_2. User a_1 knows and likes a_2's point of view, so a_1 sets manually high trust to a_2 and ranks a_2's documents d_5 and d_6 in the agent a_1 database. In this way, Personal Web of Trust $PWoT(a_1)$ includes both a_2 and the a_2's reviewers (Fig 4b). Note that the set trust regards also any document that a_2 will write in the future.

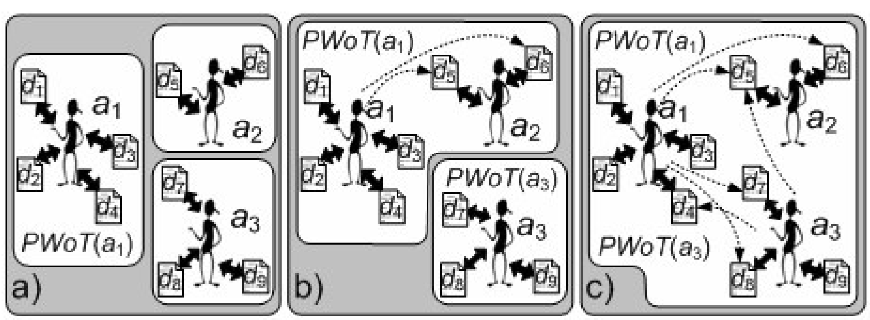

Fig. 4. The extension of $PWoT(a_1)$ of agent a_1. Thick edges denote the authorship whereas the dotted arrows – user ranks of documents

After some time, one of the reviews reminds user a_1 about the concert and another user a_3, who a_1 met there. However, after the concert a_1 lost contact with a_3. User a_1 easily finds the a_3's blog in which a post related to the concert contains a picture of both a_1 and a_3 they had taken during that event. User a_1 includes user a_3 into *PWoT*(a_1), see Fig. 4c.

Next, a_1 queries the search engine for "*Pink Floyd*" again. This time, a_1 sees reviews on the band by author a_2 at the top of the response document list. Moreover, there are more of them, e.g. a review of the Pink Floyd's video the author a_2 wrote.

Meanwhile, user a_3 after reading a_1's comment to the post on *Pink Floyd*'s concert (d_4), included a_1 and d_4 to *PWoT*(a_3). Furthermore, a_3 requests agent a_1 for trust to a_2. and rank of a_2's document d_5. User a_1 responses and a_3 adds a_2 and d_5 to a_3's Personal Web of Trust *PWoT*(a_3). Finally, *PWoT*(a_3) extends and includes the most part of *PWoT*(a_1), see Fig. 4c.

6 Conclusions and Future Work

The usage of Personal Web of Trust is another method for collaborative filtering of web documents. A user as a reader provided with a personal agent is able to process the most valuable information. An agent using user's trust to authors and ranks of documents is able to filter out and rank new documents even created by an author unknown to the reader. The filtering is based on the trust and rank exchange between co-operating agents in the multi-agent environment.

Apart from ranking of documents, the *PWoT* method may be used for tracking author's social position [10] based on the number and value of user trusts to them. This may be a valuable tool in social networks analysis [9], especially for scientific communities [17]. Another potential application of the proposed method is monitoring of knowledge dynamics based on the document's number of ranks as well as extension of various personalization systems with the collaborative trust [8, 14, 15].

One of the future improvements may be extending the method with PKI. If a user holds another user's public key certificate in the repository, it is most likely that such users share digitally signed documents or use encrypted communication and this indicates some relationship between these two users.

Besides, the semantic information about authors relationships included in documents may be used for semi-automated building of *PWoT*. The two technologies: FOAF RDF files and XFN link attributes appear to be a perfect source of semantic information if only used broader by web document authors [1, 2].

Acknowledgements

This work was partly supported by the Polish Ministry of Science and Higher Education, grant no. N516 037 31/3708.

References

1 Çelik, T., Meyer, E.: XHTML Friends Network (Poster). ACM Hypertext, 2004.
2 Dumbill, E.: XML Watch: Finding friends with XML and RDF. IBM Developer Works, 2002, http://www-128.ibm.com/developerworks/xml/library/x-foaf.html.
3. Falcone R., Singh M., Tan Y.-H.: Trust in Cyber-societies: Integrating the Human and Artificial Perspectives, Springer Verlag LNCS 2246, 2001.
4. Golbeck J., Hendler J.A.: Accuracy of Metrics for Inferring Trust and Reputation in Semantic Web-Based Social Networks. EKAW 2004, LNCS 3257, 2004, 116-131. http://www.mindswap.org/papers/GolbeckEKAW04.pdf
5 Golbeck J., Parsia B., Hendler J.: Trust Networks on the Semantic Web. Cooperative Intelligent Agents CIA 2003, LNCS 2782, Springer Verlag, 2003, 238-249.
6. Grandison T., Sloman M.: A Survey of Trust in Internet Applications. IEEE Communications Surveys and Tutorials 3 (4), 2000, http://www.doc.ic.ac.uk/~mss/Papers/Trust_Survey.pdf.
7. Guha R.V., Kumar R., Raghavan P., Tomkins A.: Propagation of trust and distrust. 13th International Conference on World Wide Web, WWW 2004, ACM Press, 2004, 403-412.
8. Kazienko P., Adamski M.: AdROSA - Adaptive Personalization of Web Advertising. Information Sciences, 2007.
9. Kazienko P., Musiał K.: Social Capital in Online Social Networks. 10th Int. Conf. on Knowledge-Based Intelligent Information & Engineering Systems, KES 2006, LNAI 4252, Springer Verlag, 2006, 417-424.
10. Kazienko P., Musiał K.: Mining Social Position of Individuals in Virtual Social Networks. AI Communication, Special Issue on Network Analysis in Natural Sciences and Engineering, 2007.
11. Khare R., Rifkin A.: Weaving a Web of trust. World Wide Web Journal, Special issue: Web security: a matter of trust, 2 (3), 1997, 77 – 112.
12. Kollingbaum M.J., Norman T.J.: Supervised interaction: creating a web of trust for contracting agents in electronic environments. The First International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS 2002, ACM Press 2002, 272-279.
13. Milgram, S.: The Small-World Problem. Psychology Today, 2, 1967, 60–67.
14. Mui L.: Computational Models of Trust and Reputation: Agents, Evolutionary Games, and Social Networks. Ph.D. Thesis. Massachusetts Institute of Technology, December 20, 2002, http://groups.csail.mit.edu/medg/ftp/lmui/computational%20models%20of%20trust%20and%20reputation.pdf.
15. O'Donovan J., Smyth B.: Trust in recommender systems. International Conference on Intelligent User Interfaces IUI 2005, ACM Press, 2005, 167-174.
16. Pujol J.M., Sangüesa R., Delgado J.: Extracting reputation in multi agent systems by means of social network topology. International Conference on Autonomous Agents, ACM Press, 2002, 467 – 474.
17. Rana O.F., Hinze A.: Trust and reputation in dynamic scientific communities. IEEE Distributed Systems Online, Vol 5, Issue 1, 2004. http://ieeexplore.ieee.org/iel5/8968/28452/01270714.pdf.
18. Richardson M., Agrawal R., Domingos P.: Trust Management for the Semantic Web. International Semantic Web Conference ISWC 2003, LNCS 2870, Springer Verlag, 2003, 351-368, http://www.cs.washington.edu/homes/mattr/doc/iswc2003/iswc2003.pdf.
19. Web of trust, Wikipedia, http://en.wikipedia.org/wiki/Web_of_trust.