

## Resource-Aware Update Policy for Highly Dynamic P2P Networks

Dragan Milosevic, Jerome Kunegis, Sahin Albayrak  
DAI-Labor, Technical University Berlin, Germany  
{dragan, jerome, sahin}@dai-lab.de

### Abstract

*As a user attention has become a precious resource, a special care has to be taken to reduce the time needed for answering queries in P2P networks. Existing solutions are usually not fully aware of the seriousness to carefully use available network resources, and they are often focused on guarantying that results are found or that given update is propagated to all peers. Since the costs for giving such guaranties might be extremely large, a core of this paper is focused on presenting one update policy that carefully utilises available network resources and tries to improve itself by learning the importance of available peers. The rationality of having such update policy is supported by analysing the usage of peers together with estimating the load values of different network links.*

**Key words:** Resource awareness, P2P Update policies, Information retrieval, Distributed filtering framework

### 1. Introduction

An exponential grow of the amount of information that is potentially online available has made that not only centralised search engines but also different P2P architectures become very attractive for IR tasks. They will first exploit sophisticated routing techniques to find information that is scattered around many distributed peers, and subsequently combine found results by utilising advanced information fusion mechanisms. To be sure that these techniques are really sophisticated and advanced, many challenges have to be addressed. Some of these great challenges arise due to a dynamic nature of real P2P networks, where peers can appear and disappear whenever they want, and where their content can change frequently often.

According to the authors' point of view, handling the frequent changes of the content that each peer can provide to others becomes even harder due to the limited network resources that can be dedicated to performing update policies. Even though it is reasonable that every peer strives at building as accurate descriptions about other peers, a current network situation may strongly discourage that as the available resources are needed for more time-critical routing or information fusion activities.

It is therefore necessary to develop one resource-aware update policy that tries to be as careful as possible while requesting that a description of a given peer should be updated, and that learns the importance of peers in order to more intelligently utilise available resources. The way how such an update policy can be developed will be the main contribution of this paper, being structure as follows. The first two sections briefly give related work and illus-

trate main problems of updating descriptions. The core of this paper is then contained in the section that present the proposed update policy in detail. A paper is finish with a section that briefly gives available evaluation results.

### 2. Related Work

A fruitful history of P2P networks has already brought many routing solutions that intelligently activate peers in order to optimise the ratio between the quality of the obtained results and the consumed network resources. Only some of them are known as power-law search, interest-based shortcuts and routing indices. The power-law search activates the neighbouring peer with the highest number of connections that is not already activated for a given query [1][2]. Interests-based shortcuts aim to directly connect peers that may be useful to each other while resolving future queries, where these direct links might be dynamically added to accommodate the changes in the underlying content [3][4]. Routing indices store what can be found in which direction and consequently manage to significantly reduce the amount of messages transferred regarding to flooding that always activates all neighbours [5].

More importantly, all these different routing strategies require some knowledge about peers, being either number of connections, found liked results or a description of the content that can be found in a particular direction. In order to work properly, every mentioned routing strategy has to keep this knowledge up-to-date, and consequently, many specialised update-policies are invented [6][7]. These update policies usually assume the propagation of a change from the peer where the change occurred to all other peers [6][8]. A great work has been done in the direction of developing algorithms that can guarantee that all peers can be optimally found [6]. Unfortunately, a broadcast seems to be a little bit unpractical even when using a good compression technique, such as those based on Bloom filters [1][9], usually because of too frequent updates in a highly dynamic Internet world.

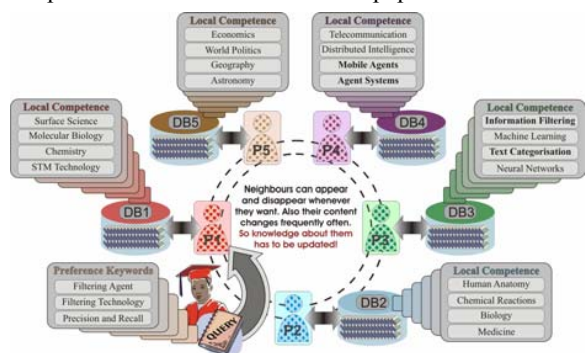
To optimise the network performances when frequent updates are necessary, authors have developed one policy [10][11] that sends the new descriptions only when being needed. Unfortunately, proposed algorithms are not taking care of limited network resources that should be carefully used especially by not very important peers in order not to slow down other ongoing activities. One step further will be to bring resource awareness by extending update policy to take care of both available network resources and the importance of peers while deciding whether to request the new description of a given peer.

### 3. Problem Description

With an exception of a routing strategy based on flooding, any other routing solution requires some descriptions of neighbouring peers in order to intelligently select those that are worth of receiving the actual query. Unfortunately, there is a great challenge of maintaining these descriptions mostly because peers are extremely dynamic, where their content may change frequently often, and also where they might appear or disappear whenever they want. Obviously, there is the necessity to realise intelligent scanning techniques which guarantee that the descriptions of peers are useful for a routing process by reason of being up-to-date.

The respecting problems, which are worth of a careful attention by any scanning approach, might be summarised as: (1) determining the importance of any peer whose description should be updated, where more important peers have rights to send their descriptions more often, (2) estimating the load of a network link that should be used for transferring the updated description, where a more loaded link means that some restrictions regarding the transmission of a description have to be made, (3) learning how often any peer changes its content in order to be able to estimate when sufficiently large changes may be received, and (4) taking care of how long ago the old description was obtained, where the age of a description increase the importance of requesting the new one.

On Fig. 1 five peers form a neighbourhood that peer P1 can use for resolving a user query. To successfully route a received query, P1 has to have a good knowledge about a competence of its neighbours. However, as its neighbours are always in larger or smaller extent dynamic, P1 has to maybe refresh the picture about their competence, being a core problem to be addressed in this paper.



**Fig. 1: Updating descriptions of peers as a challenge that has to be resolved before query routing**

The following section is going to present one approach for handling a trade-off between always having as accurate picture about peers as possible and carefully spending limited network resources on updating descriptions. The proposed approach will address fundamental problems mentioned in this section and being connected with importance and dynamics of peers, as well as with a measured load values of needed network links.

### 4. Approach

A novel authors' point of view on developing P2P filtering architecture is not only concerned with query routing and information fusion, but also with a resource-aware scanning of neighbouring peers. The challenge of deciding which peers should receive the actual query is in [11] addressed by a routing strategy that takes care of both what is known about neighbouring peers and how accurate this knowledge is. A solution for putting together results found by different peers is in [10] realised by re-ranking based on the reputation of peers. The approach proposed in this paper further extends implementations from [10][11] in a direction of making them especially suitable for the highly dynamic environments, where peers change their content frequently often. In order to make the ongoing discussions precise, the most important terms are defined as follows:

**Def. 1 (Coordination):** *Coordination* is a comprehensive activity, performed inside each peer with an ultimate goal to find how internally the received query can be processed. A way how such a coordination scheme can be achieved is out of scope of this paper, and it is addressed in [12].

**Def. 2 (Cooperation):** *Cooperation* is performed between neighbouring peers to find which ones are most competent for processing a received query. Cooperation approaches proposed in both [10][11] will be further investigated in a direction of intelligently scanning neighbours for changes.

**Def. 3 (Peer Link Load):** *Peer Link Load*  $P_L$ ,  $P_L \in [0,1]$  is the load of a network link between reference and examined peer, where the reference peer is the one that should decide whether to update the knowledge about the examined one. Larger  $P_L$  value means that link is more loaded.

**Def. 4 (Peer Dynamics):** *Peer Dynamics*  $P_D$ ,  $P_D \in [0,1]$  illustrates how often a particular peer changes its underlying content. A more dynamic content of a peer should be reflected through a larger  $P_D$  value.

An initialisation of  $P_D$  is supported by few simple heuristic rules that encapsulate domain knowledge about the dynamics of underlying document collections. It is set (i)  $P_D^{(i)}|_{t=0} = 0.75$  in the case where in a document collection of a given peer  $i$ , most of documents are the news articles known to change frequently, (ii)  $P_D^{(i)}|_{t=0} = 0.25$  when scientific papers dominate in a collection of peer  $i$ , and (iii)  $P_D^{(i)}|_{t=0} = 0.5$  in all other cases.

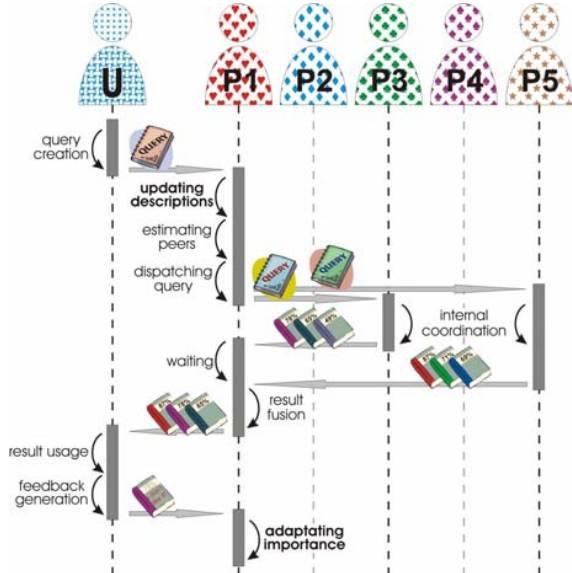
**Def. 5 (Peer Importance):** *Peer Importance*  $P_I$ ,  $P_I \in [0,1]$  shows how often a given peer has provided useful results, where the provision of more useful results gives larger  $P_I$ .

The initialisation of  $P_I$  is performed by borrowing assumptions from power-law networks [1], saying that more connected peers are more important. To always generate a value from  $[0,1]$ ,  $P_I$  is initialised as:

$$P_l^{(i)} \Big|_{t=0} = \frac{f(i)}{\max_{j \in \{1, \dots, n\}} f(j)} \quad (1)$$

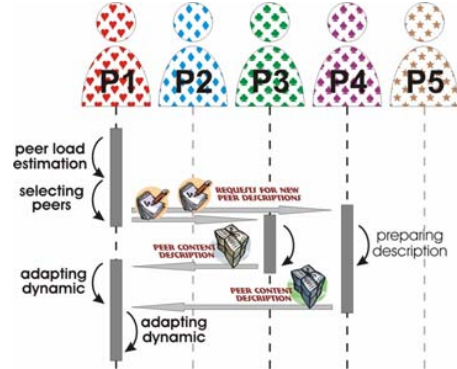
In (1), function  $f(i)$  returns number of direct neighbours that peer  $i$  has, and  $\max_{j \in \{1, \dots, n\}} f(j)$  finds a maximal known number of direct neighbours that any peer  $\{1, \dots, n\}$  might have. Under assumption that there is connection between at least two peers, it holds  $f(i) \geq 0$ ,  $\max_{j \in \{1, \dots, n\}} f(j) > 0$ ,  $f(i) \leq \max_{j \in \{1, \dots, n\}} f(j)$ , and finally  $0 \leq P_l \leq 1$ .

High level system architecture of a proposed P2P filtering architecture is given on Fig. 2. User agent (U) is responsible for the creation of queries by collecting user preferences. It also knows how user feedback can be obtained and forwarded to a chosen peer. Peer agent (P) is a cornerstone that fulfils all cooperation activities and ensures the satisfied quality of filtering services. It should be seen as an entity that is on the first place able to receive queries from user agents. On Fig. 2, U has selected P1 to be responsible for processing its query. After receiving a query, P1 first updates the obsolete content description of neighbouring peers and then estimates how each peer can be promising for processing current query. After the peer estimation, P1 is able to dispatch the received query to the right peers. On Fig. 2, P1 has found that peers P3 and P5 should be activated for a given query. As soon as the activated peers P3 and P5 have produced results, P1 will then compose the final result set that will be returned back to U. In a case of receiving any feedback from U about the relevance of results, P1 will perform the adaptation of knowledge that it has about the importance of the responsible peers. The activities that will be in the focus of this paper are on Fig. 2 highlighted by using bold font style.



**Fig. 2: System architecture that illustrates overall cooperation activities about neighbouring peers**

The process of updating the descriptions of neighbouring peers is presented on Fig. 3. It starts by estimating the load of the links between by a user chosen peer P1 and its direct neighbours. By using the determined network load values, together with the knowledge about both the importance of neighbours and their dynamics, P1 selects P3 and P4 to be asked to send their new descriptions. As soon as their new descriptions are received, they will be used for adapting the knowledge about the dynamics of P3 and P4.



**Fig. 3: System architecture that presents activities while updating descriptions of neighbouring peers**

A way how the algorithm for updating the descriptions of neighbouring peers will benefit from the introduced  $P_L$ ,  $P_D$  and  $P_I$  will be described in a rest of this paper.

#### 4.1. Peer Load Estimation

A peer link load  $P_L$  ( $0 \leq P_L \leq 1$ ) brings the network resource awareness, being crucial for postponing the process of updating descriptions of peers when the connecting link is highly loaded. It is computed as  $P_L = 1 - (1 - l_N) e^{-\frac{\beta_N t_N}{t_{to} - t_N}}$ , where  $\beta_N$  ( $\beta_N > 0$ ) is a tuning parameter,  $t_N$  as the average response time and  $l_N$  ( $0 \leq l_N \leq 1$ ) as the amount of the lost packets are obtained by the *ping* operation on the given peer whose description should be maybe updated, and  $t_{to}$  is timeout used in *ping* that thus satisfies  $t_{to} \geq t_N$ , i.e. average response time cannot be larger than a timeout.

A  $P_L$  value dependences on both  $l_N$  and  $t_N$ , where an influence of the amount of lost packets ( $l_N$ ) dominates. In the case where all sent packets are lost, or when  $l_N = 1$ , it consequently follows  $P_L = 1$ , irrespective to the value of any other parameter. This is the highest value of  $P_L$  being reasonable to be assigned in this situation where all packets are lost and where obviously network is not available. In the opposite case where all sent packets are received,  $l_N = 0$ , and the computation of  $P_L$  depends solely on average response time  $t_N$ , i.e. more loaded network results in larger  $t_N$ ,  $e^{-\frac{\beta_N t_N}{t_{to} - t_N}}$  decreases, and  $P_L$  grows, or

when  $e^{-\frac{\beta_N t_N}{t_{io}-t_N}} \rightarrow 0$  for  $t_N \rightarrow t_{io}$ , it finally holds  $P_L \rightarrow 1$ . In the intermediate case, where some sent packets are lost and some packets are received, or  $0 < l_N < 1$ ,  $l_N$  should be understood as a mean for making additional penalties on  $P_L$ . Although very small  $t_N$  will influence that  $e^{-\frac{\beta_N t_N}{t_{io}-t_N}}$  is quite close to 1,  $P_L$  is going to have small value only when the whole  $(1-l_N)e^{-\frac{\beta_N t_N}{t_{io}-t_N}}$  expression is quite close to 1. This can be achieved only when no additional penalties are paid by  $(1-l_N)$  factor, or  $l_N$  should also be small.

## 4.2. Selecting Peers

A selection process decides, based on  $P_L$ ,  $P_D$  and  $P_I$  values, which neighbouring peers should be asked to send their new descriptions. While  $P_L$  is always estimated on-the-fly as shown in a previous sub-section,  $P_D$  and  $P_I$  are first initialised as explained in the introductory part of this section, and then learned as it will be presented in the next sub-section. By using  $P_L^{(i)}$ ,  $P_D^{(i)}$  and  $P_I^{(i)}$  values that are computed for a neighbouring peer  $i$ , a decision to ask it to send its new description will be made when:

$$P_I^{(i)} P_D^{(i)} (1 - P_L^{(i)}) \geq \min(\beta_a' P_D^{(i)}, \beta_a'') e^{-\beta_D t_c} \quad (2)$$

In (2),  $\beta_a'$  ( $\beta_a' \in [0,1]$ , and usually  $0 < \beta_a' < 0.3$ ) is a so-called main activation threshold that defines how large at least a product of  $P_I^{(i)}$ ,  $P_D^{(i)}$  and  $1 - P_L^{(i)}$  should be to ask peer  $i$  to send its new description,  $\beta_a''$  ( $\beta_a'' \in [0,1]$ , and usually  $0.6 < \beta_a'' < 0.9$ ) is an auxiliary threshold ensuring that even when  $\min(\beta_a' P_D^{(i)}, \beta_a'') = \beta_a' P_D^{(i)}$  (i.e. when (2) becomes  $P_I^{(i)} (1 - P_L^{(i)}) \geq \beta_a' e^{-\beta_D t_c}$ ), peer  $i$  has chance to be sometimes activated in spite of having very low dynamics  $P_D^{(i)}$ ,  $t_c$  shows how long ago the old description was obtained, and  $\beta_D$  ( $\beta_D > 0$ ) is a suitable tuning parameter. Consequently, factor  $e^{-\beta_D t_c}$  increases a chance that old descriptions (with large  $t_c$  and thus  $e^{-\beta_D t_c} \rightarrow 0$ ) will be updated by lightening the condition (2) that  $P_L^{(i)}$ ,  $P_D^{(i)}$  and  $P_I^{(i)}$  should satisfy. More importantly, without  $\beta_a''$ , the relatively stable peers, having  $P_D^{(i)} < \beta_a'$ , will be checked for changes only if being really very old (when  $e^{-\beta_D t_c}$  significantly reduces right hand side of (2)), being maybe undesired. A better control is achieved by auxiliary threshold  $\beta_a''$ , whose values are significantly larger than for a primary threshold  $\beta_a'$ , as  $\beta_a''$  has influence only on peers that are static (very small  $P_D$ ), and which should be

checked only when being very important (large  $P_I$ ), when network conditions are promising (small  $P_L$ ), and when obtained long time ago (large  $t_c$  and thus small  $e^{-\beta_D t_c}$ ).

## 4.3. Adapting Dynamics and Importance

While the adaptation of peer dynamics  $P_D$  can be performed as soon as the new description is obtained from a requested peer, peer importance  $P_I$  can be adapted only after receiving the user feedback about the real relevance of results. The algorithms utilised for adapting  $P_D$  and  $P_I$  values will be given in the subsequent paragraphs.

The adaptation rule for learning  $P_D$  is defined as:

$$\Delta P_D = \gamma_D \min(P_D, 1 - P_D) l(t) (d_{\max} - d_J) e^{-\beta_D t_c} \quad (3)$$

In (3),  $\beta_D$  and  $\gamma_D$  are tuning parameters ( $\beta_D, \gamma_D > 0$ ), a factor  $\min(P_D, 1 - P_D)$  slows down an adaptation of  $P_D$  if reaching its extreme 0 or 1 values,  $d_J$  ( $d_J \in [0,1]$ , where larger  $d_J$  means more similar descriptions) is the Jaccard index [13] computed by comparing old and new content descriptions of a peer whose  $P_D$  is adapted,  $d_{\max}$  defines how small  $d_J$  (or how great a difference between descriptions) should be in order to increase the peer dynamics  $P_D$  value,  $l(t) = l_0 e^{-\gamma t}$  is the decreasing learning rate insuring that already learnt  $P_D$  value will not be easily destroyed, and  $t_c$  gives the amount of time passed from obtaining an old description. Therefore, factor  $e^{-\beta_D t_c}$  reduces influence of large dissimilarities between old and new descriptions (large  $d_{\max} - d_J$  value) when  $t_c$  is large, corresponding to the case when the old description is really old. In the opposite case, when the old description is recently obtained,  $t_c$  is small, and difference  $d_{\max} - d_J$  can have a significant role on adapting  $P_D$ . As far as the difference  $d_{\max} - d_J$  is concerned, always when the compared old and new descriptions are very similar,  $d_J$  has a value that is larger than  $d_{\max}$ , difference  $d_{\max} - d_J$  is negative, and finally  $P_D$  value will decrease, being reasonable as new description is almost the same as the old one. In the case where old and new descriptions differ a lot, the small value of  $d_J$  gives  $d_{\max} - d_J > 0$  and  $P_D$  is increased.

The adaptation of the peer importance  $P_I$  is based on a comparison between by a peer predicted result relevance  $q_p$  and the actual relevance  $q_a$ , being generated from a user feedback. The used adaptation rule is:

$$\Delta P_I = \gamma_I \min(P_I, 1 - P_I) l(t) (\varepsilon - |q_a - q_p|)^{2k+1} \quad (4)$$

In (4),  $\varepsilon$  is a tolerance, which defines how close the predicted relevance  $q_p$  of results should be to the actual rele-

vance  $q_a$  in order to assume that the responsible peer has found the useful results and to reward it by increasing its importance,  $k$  ( $k > 0$ ) increases the influence of large  $q_a$  deviations from  $q_p$ , a factor  $\min(P_I, 1 - P_I)$  similarly as it was the case with  $P_D$  slows down adaptation for extreme values, and the meaning of  $l(t)$  is the same as before. A reward for a good estimation  $q_p$ , being quite close to  $q_a$ , is limited to  $\frac{\gamma_l l_0}{2} \varepsilon^{2k+1}$ , being a maximal increase of  $P_I$  that can be get only when  $P_I^{(old)} = 0.5$ .

## 5. Evaluation

An implementation principle, adopted in [10][11], uses a component-based architecture, where activities on Fig. 1 are separately realised. The important advantage of such a component-based realisation lies in the possibility to easily replace any component with its new version without affecting others. Improvements proposed in this paper are thus realised by replacing a component that is responsible for updating descriptions of peers and by enhancing the adaptation component to learn the importance values.

Solutions for distributed information retrieval [10][11] have been evaluated in real conditions in the PIA system [14]. By using PIA it is possible to evaluate whether the average duration of processing queries might be reduced by saving time through efficiently updating descriptions. More importantly, a feedback from real users should show whether this reduction of response time might be obtained without sacrificing the users' satisfaction. Unfortunately, PIA is currently reserved for real-time evaluations of different collaborative filtering techniques, and the evaluation of the proposed update policy has to wait a little bit.

While waiting for the real evaluation, both analysis of available logs and measurements of network load are performed in order to strengthen an intuition behind condition (2). An analysis of available user ratings together with the information about which filtering agent has provided the rated results have shown that the small fraction of agents provided most of good results. More exactly, only 3 of 18 used filtering agents have provided 92% of all results that are rated at least by 4 on a scale from 1 to 5. As each agent can be seen as one peer, it seems that only few peers used to deliver most of good results. The proposed update policy identifies those peers as important ones (by assigning large  $P_I$  to them) and tolerates higher updating costs.

Furthermore, the measurements of the network load on different links have shown that the periods of pick loads having  $P_L \geq 0.9$  can be identified only in 8% of cases. On the other side, 76% of cases have  $P_L \leq 0.1$ . The  $P_L$  integration in (2) should thus ensure that in these 8% of cases with high loads, the update activities should be preferably omitted and should be favoured when load becomes small.

Finally, evaluation results published in [11] show that the expected period of updating descriptions clearly depends on the type of underlying documents, where descriptions for news and scientific domains should be updated every 2 and 10 hours, respectively. The proposed initialisation and adaptation of  $P_D$  ensure that news domain has smaller  $P_D$  than the scientific one. The update policy, realised by (2), thus gives priority to news domain, being more dynamic.

## 6. Conclusion

A goal of this paper was to give solutions to challenges of updating descriptions of peers when available network resources are very limited. This is achieved by proposing a resource-aware update policy that learns importance of peers to be able to carefully spend available resources. A proposed update policy is realised as a component whose usefulness is speculatively estimated by analysing weblogs and measuring load values of network links. The future work will be concentrated on a real evaluation that should check whether a possible reduction of time needed for delivering results can be obtained without reducing the users' satisfaction. This evaluation will provide concrete evidences about the efficiency of a learning scheme proposed for finding right importance and dynamics of peers.

## 7. References

- [1] Tang C., *Data Sharing and Information Retrieval in Heterogeneous Distributed Systems*, Ph. D., Univ. Rochester, 2004.
- [2] Botros S., Waterhouse S., *Search in JXTA and Other Distributed Networks*, P2P 2001, Linköping, Sweden, 30-35, 2001.
- [3] Syed A., Pang X., *Knowledge Sharing Over P2P Knowledge Networks: A Peer Ontology And Semantic Overlay Driven Approach*, iKMS 2002, Singapore, 329-339, 2004.
- [4] Sripanidkulchai K., Maggs B., Zhang H., *Enabling Efficient Content Location and Retrieval in Peer-to-Peer Systems by Exploiting Locality in Interests*, CC Review, 32(1), p. 80, 2002.
- [5] Crespo A., Garcia-Molina H., *Routing Indices for Peer-to-Peer Systems*, ICDCS 2002, Vienna, Austria, 23-24, 2002.
- [6] Franconi E., Kuper G., Lopatenko A., Zaihrayeu I., *A Distributed Algorithm for Robust Data Sharing and Updates in P2P Database Networks*, EDBT 2004, Greece, 446-455, 2004.
- [7] Franconi E., Kuper G., Lopatenko A., Zaihrayeu I., *Queries and Updates in the coDB Peer to Peer Database System*, 30<sup>th</sup> VLDB Conf., Toronto, 1277-1280, 2004.
- [8] Madhavan J., Halevy A., *Composing Mappings Among Data Sources*, VLDB 2003, Berlin, 572-583, 2003.
- [9] Rhea S., Kubiawicz J., *Probabilistic Location and Routing*, INFOCOM 2002, New York, USA, 2002.
- [10] Albayrak S., Milosevic D., *Distributed Community Cooperation in Multi Agent Filtering Framework*, ICEIS 05, 400-407
- [11] Albayrak S., Milosevic D., *Cooperative Community Selection in Multi Agent Filtering Framework*, IAT 05, 527-535, 2005.
- [12] Albayrak S., Milosevic D., *Self Improving Coordination in Multi Agent Filtering Framework*, IAT 04, 239-245, 2004.
- [13] Han J., Kamber M., *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers, 2001.
- [14] Albayrak S., Wollny S., Lommatzsch A., Milosevic D., *Agent Technology for Personalized Information Filtering: The PIA-System*, SCPE, (to appear)