Self-Organization of a Small World by Topic

Christoph Schmitz

FG Wissensverarbeitung, FB 17, Universität Kassel, 34121 Kassel, Germany
schmitz@cs.uni-kassel.de

Abstract. In this paper, we demonstrate how small worlds – i. e., highly clustered networks with small diameter – of peers can organize themselves in an ontology-based P2P knowledge management system. To that end, we provide rewiring strategies to build a network in which peers form clusters where they are connected to other peers interested in similar topics. These strategies only rely on the local knowledge of each peer and on the notion of similarity derived from the relationships of entities the ontology. We evaluate these algorithms and show that a small world of topically coherent clusters actually emerges, and that this improves the performance of query operations.

1 Introduction

In many networks found in nature and society, a so-called *small world* structure has been observed; these networks exhibit special properties, namely, a small average diameter and a high degree of clustering, which make them effective and efficient in terms of spreading and finding information.

The main focus of this paper is to examine the use of small world topologies in an ontology-based P2P knowledge management (P2PKM) system. We assume that in such a system, each peer maintains a knowledge base describing a part of the world relevant to its user. These knowledge bases can then be queried locally or by other peers in the network.

In this paper, we show a way of letting peers in a P2PKM setting organize themselves into a small world structured around the topics that peers contain knowledge about, and how this small world topology can improve the performance of query routing strategies.

The remainder of this paper is structured as follows: Sections 2 and 3 introduce related work and the necessary terms and definitions, respectively. Routing and rewiring algorithms to build a small-world semantic P2P network are described in Section 4 and evaluated in Section 5. Section 6 provides a conclusion, an interpretation of the results and outlook.

2 Related Work

Recently, there have been several research projects concerned with knowledge management in a P2P setting; examples include Edutella [1] and SWAP [2]. The

assumptions about P2PKM systems made in Section 3 are coherent with what has been done in these projects.

Much of this paper draws upon the observations on the structure and evolution of small-world networks presented by Barabási [3] and Watts/Strogatz [4]. Barabási demonstrates how the structure of existing networks, such as the hyperlink topology of the WWW, can be replicated by a process of *preferential attachment*, meaning that each node in the network will be linked to by others with a probability proportional to its number of links.

Watts and Strogatz [4] describe the basic notions of the *clustering coefficient* and *characteristic path length* measures as indicators of small-world networks. Watts [5] examines several models for the growth of small-world networks from what he calls *substrates*: e. g. a substrate could be a ring or a grid graph from which a small-world topology emerges. He introduces the idea of *rewiring* as a mechanism for evolving the graph structure of a substrate into a more desirable one.

Both Barabási and Watts/Strogatz, however, assume a *global view* on the graph; for example, to attach preferentially to a node with high indegree in the Barabási model, a new node would have to see all nodes in the graph in order to assess the respective degrees. In Section 4, we show strategies for rewiring into a small-world network which only rely upon the local knowledge present at each node.

The *knows* relation presented in Section 3 is a variant of the *routing index* as presented by Crespo et. al. [6], extended from a keyword-based version into one that contains arbitrary items comparable by a distance function.

Clustering is mentioned as an enabling factor for routing strategies such as *fireworks routing* [7] or the superpeer-based routing in Edutella [8]. We complement these works by presenting a strategy for building the necessary clustered structure and demonstrate under which circumstances clustering is beneficial for routing. [7] mentions a *Learning Fuzzy* method for topology building (which is not elaborated in the paper), which may be similar to the rewiring strategies mentioned in Section 4.

Haase and Siebes [9] describe P2P routing strategies based on semantic topologies. In contrast to this paper, they use an approach of pushing advertisements of one's own expertise to other peers, as opposed to the rewiring strategies used here which pull information about suitable new neighbors from the network. Furthermore, they do not make any explicit observations about the graph structure of the emergent network.

3 Basics and Definitions

3.1 Model of the P2P network

As this paper is mainly concerned with network topologies and routing strategies in P2PKM systems, we abstract from the details of a P2PKM system implementation and make the following assumptions (similar to [9]):

– Each peer stores a set of *content items*. On these content items, there exists a *similarity function sim* which can be used to determine the similarity of content items to each other. We assume $1 - sim$ to be a metric.
– Each peer provides a self-description of what it contains, in the following referred to as *expertise*. Expertises need to be much smaller than the knowledge bases they describe, as they are transmitted over the network and used in other peers' routing tables. In our case, the expertise consists of a content item selected as representative for the peer, but in general, the expertise could also include peer metadata like query languages supported, additional capabilities of the peer etc. As peer expertises are content items, they can be compared to each other and to queries using the *sim* function.
– There is a relation *knows* on the set of peers. Each peer knows about a certain set of other peers, i. e., it knows their expertises and network address (IP, JXTA ID). This corresponds to the routing index as proposed by Crespo et al. [6]. In order to account for the limited amount of memory and processing power, the size of the routing index at each peer is limited.
Sometimes it is more convenient to talk about the network in the terms of graph theory. One can view the P2P network as a directed graph $G(V, E)$ with a set of *nodes* V and a set of *edges* $E \subseteq V \times V$, where each peer P constitutes a node in V, and $(P_1, P_2) \in E$ iff $knows(P_1, P_2)$. We will use both notations synonymously.
– Peers query for content items on other peers by sending query messages to some or all of their neighbors; these queries are forwarded by peers according to some *query routing strategy*. Using the *sim* function mentioned above, queries can thus be compared to content items and to peers' expertises.

3.2 (Weighted) Clustering Coefficient

One observation about small-world networks found in many areas such as sociology or biology is that there are *clusters* of nodes. This means, loosely speaking, that for each node, its neighbors are likely to be connected directly themselves.

More specifically, the clustering coefficient for a node v has been defined [5] as the fraction of possible edges in the neighborhood of a node which are actually present. We slightly modify that definition to use a directed graph as our *knows* relation may be asymmetric.

$$\gamma_v = \frac{1}{k_v(k_v - 1)} \sum_{w \in \Gamma(v)} |\{u \in \Gamma(v) : (w, u) \in E\}| \tag{1}$$

where $\Gamma(v)$ are the nodes pointed to by v, not including v:

$$\Gamma(v) = \{u \in V \backslash \{v\} : (u, v) \in E\} \tag{2}$$

and $k_v = |\Gamma(v)|$ is the size of the neighborhood. As $k_v(k_v - 1)$ in Equation 1 is the maximum number of edges possible in the neighborhood, γ_v takes on values between 0 and 1.

The clustering coefficient $\gamma(G)$ of a graph is the mean of the clustering coefficient over all nodes.

In the following, we extend this notion to a *weighted clustering coefficient* γ^w. The motivation for this is that we do not only want to capture how densely connected the neighborhood of each peer is, but also if the neighbors have contents similar to that of the respective peer:

$$\gamma_v^w = \frac{1}{k_v(k_v-1)} \sum_{w \in \Gamma(v)} sim(v,w) \, |\{u \in \Gamma(v) : (w,u) \in E\}| \tag{3}$$

This means that for the weighted clustering coefficient of node v, each edge from a neigbor w counts only as much as the similarity between w and v.

The weighted clustering coefficient is related to the observation that in actual small-world networks where there is a notion of similarity between nodes, nodes are not only surrounded by dense neighborhoods, but the neighboring nodes tend to be similar to the node under consideration. In a social network of humans, for example, you are likely to find people of common interests in these clusters.

With the above definitions, we have $0 \leq \gamma_v^w \leq 1$. Large values of γ_v^w mean that v is surrounded by a *dense* neighborhood of *similar* nodes.

3.3 Characteristic Path Length

The characteristic path length is a measure for the mean distance between nodes in the network. It is defined by Watts [5, p. 29] as follows: "The *characteristic path length* (L) of a graph is the *median* of the *means* of the *shortest path lengths* connecting each vertex $v \in F(G)$ to all other vertices. That is, calculate $d(v,j)$ $\forall j \in V(G)$ and find $\overline{d}_v$ for each v. Then define L as the median of $\{\overline{d}_v\}$." Here, $d(v,j)$ is the number of edges on the shortest path from v to j, and $\overline{d}_v$ is the average of $d(v,j)$ over all $j \in (V - \{v\})$.

For reasons of efficiency, we use the sampling technique proposed by Watts (take a sample $\{v_1, \ldots, v_m\} \subset V$ for some $m < |V|$, compute mean distance $\overline{d}_{v_i}$ for each, take median of mean distances as CPL) to estimate L. Note again that in contrast to [5], we consider our network to be directed.

3.4 Ontologies

We assume that in the P2P knowledge management system, peers operate on knowledge represented in terms of *ontologies*. More specifically, in this paper we will use the terms "ontology" and "knowledge base" as defined in the context of the KAON ontology framework; for details refer to [10].

In short, a *core ontology* consists of a partially ordered set of *concepts*[1], the partial order being "subconcept of", and *relations*[1] between these concepts. For example, there could be concepts *Professor* and *PhDStudent*, and a relation *supervises(Professor, PhDStudent)* between them. A *knowledge base* consists of a

[1] More precisely: *concept identifiers* and *relation identifiers*; we will stick to the simplified terminology of [10] here.

core ontology plus instances of the concepts and relations; e. g. a knowledge base using the above concepts could contain *stumme* as an instance of *Professor*, *schmitz* as an instance of *PhDStudent*, and *supervises(stumme, schmitz)* instantiating the supervises relation.

4 Rewiring and Routing Algorithms

In a social system, people tend to be surrounded mostly by people who are similar to themselves in some sense. A librarian will relate to other people who care about books, and a surgeon will probably know some more people from the health care area. This leads to the following observation: if I want to find out something about, say, a possible cure for squinting, I may want to ask my friend, the surgeon. Even though he possibly does not know very much about squinting himself, chances are that he will know an ophthalmologist. On the other hand, if people are related to very similar people only, this will lead to so-called "caveman worlds" [5], i. e. disconnected cliques which are not connected to each other. In practice, however, many people maintain relationships to people from different professions, geographical locations, etc.; these are called *long range edges*.

To apply these observations in a peer-to-peer setting and allow our P2P network to organize itself to mimic the behavior of a social system, we need algorithms to make sure that peers can move within the network by establishing new connections and abandoning old ones, trying to get into clusters of similar peers. Peers also need to be able to estimate which of their outgoing edges are most suitable for forwarding an incoming query.

Figures 1 and 2 show examples of clustered and unclustered networks, respectively[2]; both have been laid out using a spring embedder algorithm. While in Figure 1 the nodes are linked randomly, in Figure 2, a topical structure can be observed. In the top left, there are peers concerned with persons and research projects (Lecturer, Professor, etc.), in the middle, there are peers containing entities related to organizations, and in the bottom right corner peers are clustered which deal with publications. In the latter of these two networks, it is reasonable to assume that peers can make "educated guesses" about the best direction in which to forward any given query, as most peers which will be able to answer can be found in a limited region of the network. In the following paragraphs, we introduce two simple strategies to cluster P2P networks by topic.

4.1 Rewiring Algorithms

In order to build a topically clustered graph using only the kind of knowledge available locally on the peers, we use strategies based on *walks* on the P2P network. To become part of a topically clustered neighborhood, i. e. to be surrounded by similar peers, a peer P_k will periodically initiate the following procedure:

[2] In this case, for the sake of simplicity, each peer contains exactly one label from an ontology. See Section 5.1 for details.

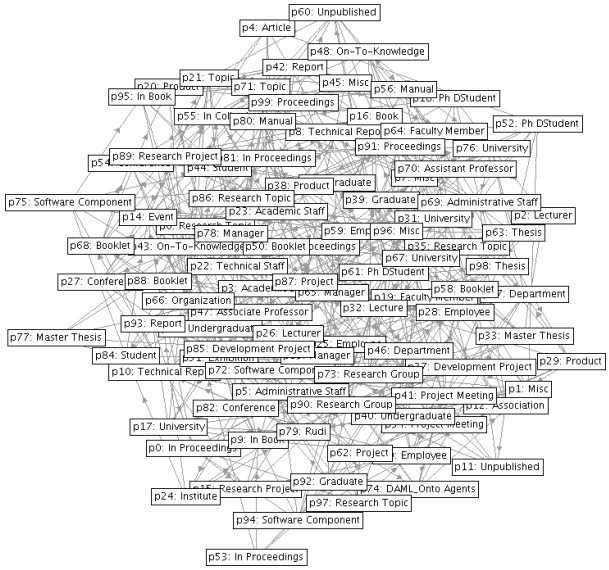

Fig. 1. Unclustered network

1. P_k assesses its *knows* relation and decides whether it is in an unsuitable neighborhood, i. e. on the average, its neighbors are too dissimilar from itself:

$$\frac{1}{k_{P_k}} \sum_{P_j \in \{P_j | knows(P_k, P_j)\}} sim(P_k, P_j) < minSimilarity \tag{4}$$

2. If so, P_k sends a *WalkMessage* M, containing its expertise and a time-to-live (TTL) value, into the network.
3. Message M is forwarded until $TTL = 0$; each forwarding peer appends its own expertise to M.
4. If $TTL = 0$, M is sent back to the original sender P_k.
5. P_k collects the other peers' expertises from M. It may find one or more suitable neighbors in that set and decide to keep these in its own routing index. If the routing index size is exceeded, entries for other – less similar – peers may have to be dropped.

The forwarding in step 3 can be done in different ways:

Random Walk. The message M is forwarded randomly. This is the best one can do if the network is not clustered yet.

Gradient Walk. At each peer P_i, the message M is forwarded to the neighbor of P_i which is most similar to the original sender P_k of M. This is suitable if the network already has a structure corresponding to the ontology (as shown in Figure 2); in a random network, however, this strategy will get stuck in local minima too easily.

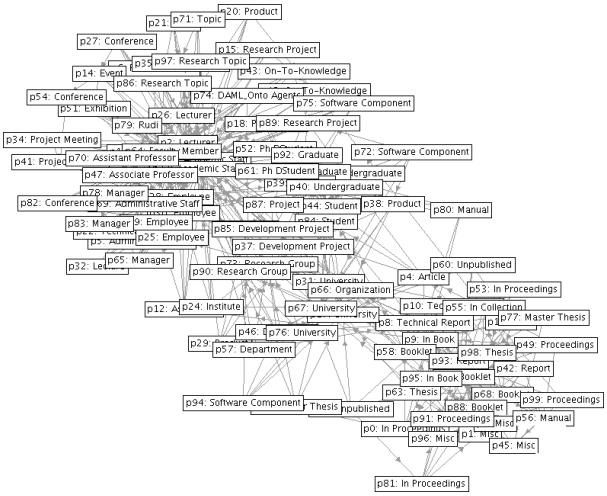

Fig. 2. Clustered network

4.2 Routing Strategies

We have experimented with a number of routing strategies which promise to be useful under the assumptions we made in the preceding sections:

Fixed Fanout Forwarding. The query is forwarded to a fixed number n of neighbors; these are selected to be the n neighbors most similar to the query.
Threshold Forwarding. The query is forwarded to all neighbors which are more similar to the query than a given threshold.
Fireworks. If the query is more similar to the expertise of the forwarding peer than a given threshold, it is broadcast in the neighborhood of the forwarding peer with a new TTL.
Fixed Fanout Random Forwarding. The query is forwarded to a fixed number of randomly selected neighbors.
Random Composite Strategy. A meta-strategy which wraps a number of other strategies plus corresponding weights, and hands over each query to one of the wrapped strategies which has been selected randomly according to the weights. For example, if we wrap strategies A and B with weights 2 and 1, respectively, A will get to handle twice as many queries as B.
Composite Strategy. A meta-strategy using a *chain of responsibility* [11] of strategies; each strategy can claim that it has processed the query, or pass it to the next strategy in the chain. Figure 3 shows an example of a composite strategy. First the query is processed locally. Then, the Fireworks strategy gets to handle it, broadcasting it if the necessary level of similarity is met. Otherwise, the query is handled by the Random Composite Strategy, which randomly chooses to hand it over to the Fixed Fanout Strategy.

This way, combinations of different routing strategies can be assembled flexibly. In practice, this will be done by an expert who designs and implements the actual P2PKM system. One can also imagine a P2PKM system learning appropriate strategies over time (cf. outlook in Section 6.2).

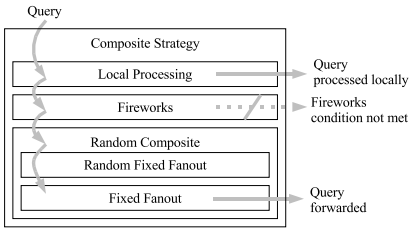

Fig. 3. Chained routing strategies

5 Evaluation

5.1 Setting

As actual P2P networks cannot easily be instantiated with arbitrary numbers of peers, different routing strategies etc., we conducted a number of experiments in a simulation environment. In our experiments, we used 500 peers, each of which was allowed to have a routing index of size 10. Each peer was assigned one randomly chosen item from the SWRC ontology[3], which acted both as content and as expertise.

Rada [12] argues that the graph distance, i. e. the number of edges between two entities in a semantic structure such as an ontology, is a valid metric of the similarity of entities. Thus, the metric in our experiment was defined by shortest paths in the graph of the SWRC ontology. Non-taxonomic relationships (such as *cooperatesWith*) were given twice the length of taxonomic relationships such as *instanceOf* or *subclassOf*. This means that taxonomic relationships bind entities tighter that non-taxonomic ones. The distances between pairs of entities were then divided by their average distance from the root concept; the motivation for this is that two sibling concepts further down the hierarchy, say, *Lecturer* and *Professor*, would be regarded as more closely related than sibling concepts at the top of the hierarchy, such as *Topic* and *Person*.

The substrate each experiment started with was as a 10-regular random graph, i. e. the *knows* relation of each peer was initialized with the 10 randomly selected other peers. If not stated otherwise, the following parameters were used:

- fireworks strategy with broadcast threshold of 1 (meaning, broadcast only if query matches peer exactly), broadcast TTL of 2
- if the fireworks strategy does not handle the query, it is forwarded to the two best matching neighbors
- the query TTL was set to 5
- the minSimilarity was set to 0.7
- composite rewiring strategy which randomly chooses between random walk or gradient walk rewiring with equal probability, both with a TTL of 5
- the starts of the rewiring processes at the peers were uniformly distributed over the time interval $[8000, 12000]$; the time between invocations of the rewiring processes was randomly selected from a normal distribution of 500 with a standard deviation of 100.

[3] `http://ontobroker.semanticweb.org/ontos/swrc.html`

5.2 Clustering Coefficients

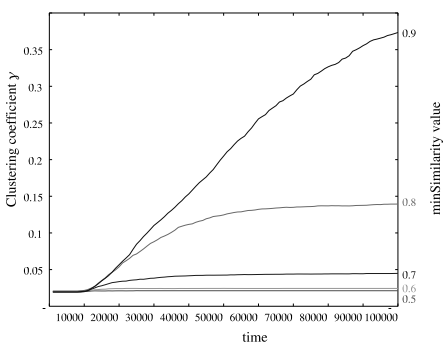

Fig. 4. Clustering coefficients over time, for different *minSimilarity* values

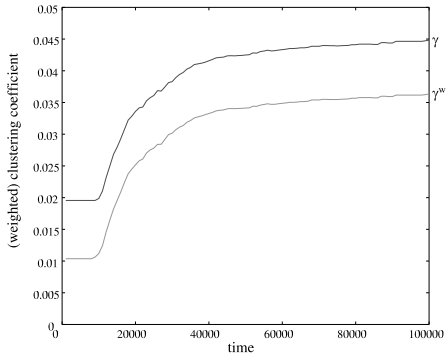

Fig. 5. (Weighted) clustering coefficient for $minSimilarity = 0.7$

Figures 4 and 5 show that the clustering coefficients γ increase as intended as a result of the rewiring process. The influence of the *minSimilarity* parameter can be seen: the higher the demands of the rewiring peers are as to the *minSimilarity* of their neighborhoods, the more the clustering coefficients increase. The same holds for the weighted clustering coefficient γ^w, proving that we are building a network of topically related clusters.

Furthermore, note in Figure 5 that the weighted clustering coefficient γ^w increases much more than the clustering coefficient (a factor of 3.5 vs. 2.2), relative to the values at the beginning. This is an indication that the clusters that are formed actually consist of topically related peers.

5.3 The Influence of Clustering on Recall and Network Load

As can be seen in Figure 6, the recall of the queries sent by the peers increases as a result of the rewiring – except for the extreme case with $minSimilarity = 0.9$, which will be discussed in Section 5.4. In the best case for $minSimilarity = 0.7$, an increase of 44% (0.27 vs. 0.39 recall) could be achieved.

At the same time, the number of messages needed per result decreases. Figure 7 shows the ratio between the number of messages needed to process each query (query and response messages) and the number of items retrieved. At the end of the rewiring process, about 9 messages are needed to retrieve one result, as opposed to 14 at the beginning (a 36% decrease).

5.4 Clustering Too Much

While the previous section has shown that a certain amount of clustering is beneficial for query routing, it is possible to cluster too much. Figure 6 shows the recall for different values of the *minSimilarity* parameter (and thus, different amounts of clustering as shown in Figure 4). We can see that for average *minSimilarity* values of around 0.7, the level of clustering achieved is optimal.

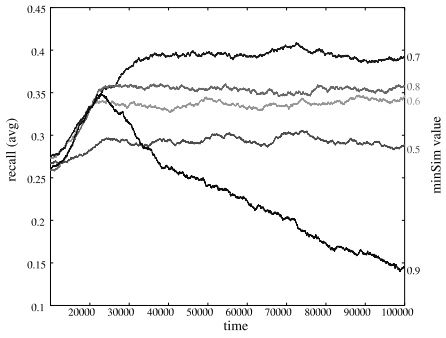

Fig. 6. Recall over time, averaged over 10000 timesteps

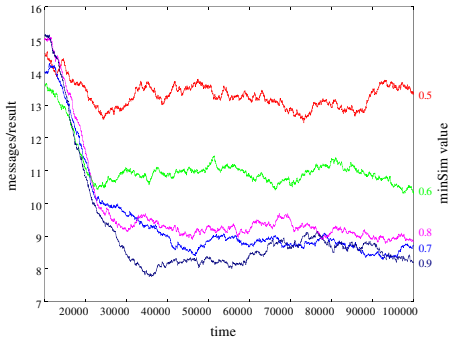

Fig. 7. Messages per result obtained, averaged over 10000 timesteps

Lower values don't yield much clustering and improvement in the querying performance at all, while higher values tend to produce clusters which are too tight, thus sacrificing inter-cluster connections. This can lead to "caveman worlds", where each cluster (cavemen in one cave) is very dense, with next to no connections to the outside world. For values of *minSimilarity* close to 1, the graph will even be partitioned into unconnected components.

5.5 Characteristic Path Length

As we started from a regular random graph, the characteristic path length (CPL) of the network was quite small from the outset [5]. Although a high clustering coefficient and small CPL are in many cases contradicting goals (e. g. a hypercube or a random graph have small CPLs, but also small clustering coefficients), Figure 8 shows that the characteristic path length was not increased much in the rewiring process, while the clustering coefficient increased (see Section 5.2). For the case yielding the best recall – namely, $minSimilarity = 0.7$ – the CPL increased only about 2%. For a higher *minSimilarity* value, the CPL increase was larger. This is an indication that in that case we cluster too much (cf. Section 5.4).

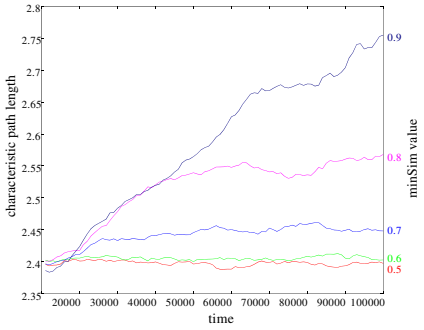

Fig. 8. Characteristic path length over time for different *minSimilarity* values

6 Conclusion and Outlook

6.1 Conclusion

In this paper, we have demonstrated how peers in an ontology-based P2P knowledge management scenario can organize themselves into a network topology which reflects the structure of the ontology – i. e., peers having similar contents

get to be close to each other in the network, thus forming clusters around common topics. We have provided simple algorithms which can be executed on each peer without central control to create this kind of topology, and have shown that a clustered topology is beneficial for query routing performance. We have also demonstrated that clustering can be overdone, yielding poorer query results.

Furthermore, we have introduced the notion of a *weighted clustering coefficient* to measure if the clusters that are forming relate to common topics, and provided interpretation of the routing performance with respect to the graph structure of the emerging network.

6.2 Outlook and Future Work

- In Section 5, we used a simplified setting where each peer contained just one entity out of an ontology. In real-world scenarios, each peer would contain a complete knowledge base. Therefore, we need to address two open questions:
  1. How can we aggregate the knowledge base of each peer into a reasonably small expertise? The expertise must express the essence of what the knowledge base is about, but at the same time be small enough to be efficiently used in making routing decisions. We are currently investigating the use of graph clustering techniques on knowledge bases in order to partition them into meaningful clusters. From these clusters, representatives for the content of the knowledge base can be selected.
  2. If we have an expertise of more than just one content item: how can we adapt the similarity function in order to be able to compare expertises against each other or against queries?
- The methods described in this paper work well if one assumes that all peers share the same ontology (at least, parts of one ontology large enough so that the shared entities can be used to do the routing). This may or may not be realistic, depending on the kind of community which wants to use this kind of semantic P2P network. If the community was not a closely-knit one which can easily agree on some standard ontology for the expected KM tasks, the problems of emergent ontologies, ontology alignment and mapping, standard upper ontologies etc. apply and would have to be solved.
  Multiple groups of users, each of which agrees on a standard ontology, would be quite easy to accommodate in a network as described here, however. The use of a certain ontology could be incorporated into each peer's expertise and thus be considered in the routing process.
- Our framework for routing and clustering in P2PKM leaves a lot of room for tuning parameters and combinations of strategies. In an implementation of such a network for end-users, one would need to hide all of these parameters and either find default values suitable for a wide range of possible network states, or find ways for each peer to automatically determine reasonable values, thus enabling the network as a whole to learn suitable parameters.
  Furthermore, it is worth discussing which of the parameters sould be user-settable. It could be necessary to limit users' possibilities in order to prevent

users from accidentally or malevolently flooding the network with query messages; on the other hand, a user should be able to express her preferences, e. g. to trade off time against precision.
- The clusters in the network can be seen as *communities* of peers with common interests. Making these communities explicit would facilitate tasks such as browsing: if a user finds that peer P_k contains interesting material, she might want to browse the contents of other peers in the community of P_k. Complementary to querying, browsing would provide a different way of accessing the knowledge available in the network. Maintaining and labeling such communities in a decentralized manner would be an interesting extension of P2P knowledge management systems.

Acknowledgement: Part of this work was funded by the German Federal Ministry of Education and Research (BMBF) in the PADLR project.

References

1. Nejdl, W., Wolf, B., et al.: EDUTELLA: A P2P networking infrastructure based on RDF. In: Proc. 11th International World Wide Web Conference. (2002)
2. Tempich, C., et al.: SWAP: Ontology-based knowledge management with peer-to-peer. In: Workshop ontologiebasiertes Wissensmanagement, WM 2003, Luzern, Bonn, Gesellschaft für Informatik (2003)
3. Barabási, A.L.: Linked: How Everything Is Connected to Everything Else and What It Means. Plume (2003)
4. Watts, D.J., Strogatz, S.: Collective dynamics of 'small-world' networks. Nature **393** (1998) 440–442
5. Watts, D.J.: Small Worlds – The Dynamics of Networks between Order and Randomneess. Princeton University Press, Princeton, New Jersey (1999)
6. Crespo, A., Garcia-Molina, H.: Routing indices for peer-to-peer systems. In: Proceedings of the International Conference on Distributed Computing Systems (ICDCS), Vienna, Austria (2002)
7. Hang, N.C., Cheung, S.K.: Peer clustering and firework query model. In: Proceedings of the 11th International World Wide Web Conference. (2002)
8. Nejdl, W., Wolpers, M., Siberski, W., Schmitz, C., Schlosser, M., Brunkhorst, I., Löser, A.: Super-peer-based routing and clustering strategies for RDF-based peer-to-peer networks. In: Proc. 12th International World Wide Web Conference, Budapest (2003)
9. Haase, P., Siebes, R.: Peer selection in peer-to-peer networks with semantic topologies. In: Proc. 13th International World Wide Web Conference, New York City, NY, USA (2004)
10. Stumme, G.: Using Ontologies and Formal Concept Analysis for Organizing Business Knowledge. In: Wissensmanagement mit Referenzmodellen – Konzepte für die Anwendungssystem- und Organisationsgestaltung. Physica, Heidelberg (2002)
11. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns – Elements of Reusable Object-Oriented Software. Addison-Wesley (1995)
12. Rada, R., Mili, H., Bicknell, E., Blettner, M.: Development and application of a metric on semantic nets. IEEE Transactions on Systems, Man and Cybernetics **19** (1989) 17–30