Information Integration in Schema-Based Peer-To-Peer Networks

Alexander Löser[1], Wolf Siberski, Martin Wolpers, and Wolfgang Nejdl[2]

[1] CIS, Technical University Berlin, 10587 Berlin, Germany
aloeser@cs.tu-berlin.de
[2] Learning Lab Lower Saxony, University of Hannover, 30167 Hannover, Germany
{siberski,wolpers,nejdl}@learninglab.de

Abstract. Peer-to-peer (P2P) networks have become an important infrastructure during the last years. Using P2P networks for distributed information systems allows us to shift the focus from centrally organized to distributed information systems where all peers can provide and have access to information.
In previous papers, we have described an RDF-based P2P infrastructure called Edutella which is a specific example of a more advanced approach to P2P networks called schema-based peer-to-peer networks. Schema-based P2P networks have a number of advantages compared with simpler P2P networks such as Napster or Gnutella. Instead of prescribing one global schema to describe content, they support arbitrary metadata schemas and ontologies (crucial for the Semantic Web). Thereby they allow complex and extendable descriptions of resources thus introducing dynamic behavior to the former fixed and limited descriptions, and can provide complex query facilities against these metadata instead of simple keyword-based searches.
In this paper we will elaborate topologies, indices and query routing strategies for efficient query distribution in such networks. Our work is based on the concept of super-peer networks which provide better scalability compared to traditional P2P networks. By adapting existing concepts of mediator-based information systems to super-peer based networks, as we will show in this paper, they are able to support sophisticated routing, clustering and mediation strategies based on the metadata schemas and attributes. The resulting routing indices can be built using local clustering policies and support local mediation and transformation rules between heterogeneous schemas, and we sketch some first ideas for implementing these advanced functionalities as well.

1 Introduction

Recently information systems that use peer-to-peer (P2P) networks as infrastructure evolved from simple P2P-based systems like Napster and Gnutella to more sophisticated ones based on distributed indices (e.g. distributed hash tables) such as CAN [19] and CHORD [21]). Using P2P-topologies for information systems enables us to shift the focus from centrally organized to distributed information systems where all peers can provide and have access to information in the network. These approaches provide the advantages of P2P topologies, e.g. robustness and flexibility. At the same time some new problems arise, e.g. fast and reliable data retrieval or efficient search. In this paper we

J. Eder and M. Missikoff (Eds.): CAiSE 2003, LNCS 2681, pp. 258–272, 2003.

address some of the problems associated with web content management and distribution, focusing on the handling of complex metadata[1] sets for data description and the support of complex queries for data retrieval.

We assume that these queries are expressed based on the schemas used for annotation. In order to query only those peers capable of answering we obviously must investigate more advanced routing algorithms than simple query broadcast. Therefore, based on information about schemas used by each peer, we create and maintain explicit query routing indices which facilitates more sophisticated routing approaches. The query is still evaluated by the peers holding the metadata sets, but only peers having annotations based on the schema elements used in the query will receive it. The routing indices do not rely on a single schema but can contain information about arbitrary schemas used in the network.

Allocating these routing indices together with clustering and mediation functionality at every peer would require a considerable amount of processing power at each peer. Also, because peers tend to join and leave the network unpredictably, the topology would be subject to constant inefficient reorganization. Therefore, we use a super-peer topology for these schema-based networks, where designated super-peers with high availability, processing power and bandwidth form a network backbone, and each peer connects to one super-peer only (see [25] for the general characteristics of super-peer networks; Kazaa, Grokster and Morpheus are examples of such super-peer systems). The super-peers are responsible for construction and maintenance of routing indices and for query routing. To support reorganization within the network each super-peer uses a so called clustering-policy. Such a policy constrains the set of peers accepted by a particular super-peer. For example, a super-peer may use a policy to accept only peers which use the Dublin Core schema. We use these policies 1) to induce network clustering based on content with the goal of reducing the amount of query broadcast and 2) to restrict the set of schemas for a particular super-peer. Restricting schemas allow us to define local mapping rules -correspondences- between schemas of a particular super-peer. Since clustering rules restrict the amount of schemas and attributes for each super-peer, we introduce a global schema at each super-peer and map peer schemas to it. We show how such a mapping is done within a particular super-peer.

There are only a few research groups that have investigated these schema-based P2P networks so far. In our group we have been working on a schema-based network called Edutella [15][16] (see http://edutella.jxta.org for the source code), which aims at providing access to distributed collections of digital resources through a P2P network. Resources in the Edutella network are not described using ad hoc metadata fields (like Napster & Co), but use RDF schemas and RDF metadata for their description. In order to retrieve information stored on the Edutella network we use the query language RDF-QEL. RDF-QEL is based on Datalog semantics and thus compatible with all existing query languages, supporting query functionalities which extend the usual relationally complete query languages.

Two other interesting approaches are the ones investigated by Bernstein et al. and Aberer et al. Bernstein et.al. [5] propose the Local Relational Model (LRM) enabling general queries to be translated into local queries with respect to the schema supported

[1] We use the terms metadata and annotations synonymously.

at the respective peer, using the concept of local translation/coordination formulas to translate between different schemas. Aberer et.al. [2,1] propose schema-based peers and local translations to accommodate more sophisticated information providers connected by a Gnutella-like P2P topology.

In section 2 we will describe the general topology of our schema-based super-peer network and the indices used to route queries. We will then discuss clustering and mediation algorithms in such networks in section 3.

2 Schema-Based Routing in P2P Networks

P2P networks that broadcast all queries to all peers don't scale. We therefore propose a super-peer topology for these networks and the use of indices at these super-peers to address scalability requirements. These indices are built using schema information from their associated peers. The super-peer network constitutes the "backbone" of the P2P network which takes care of message routing and integration / mediation of metadata.

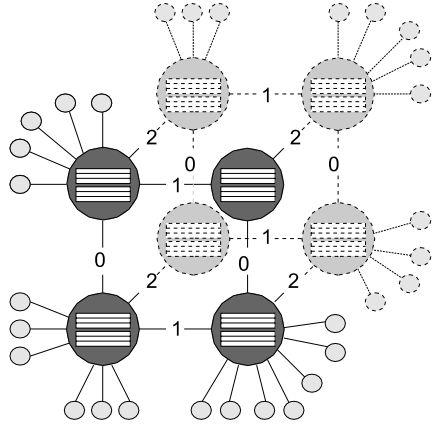

Fig. 1. Peers connected to the super-peer "backbone"

We will assume that the super-peers in our network are arranged in the HyperCuP topology [20]. Most solutions we propose in this paper could be realized with other super-peer topologies, too, which would actually lead to interesting extensions derived from the ideas in this paper. We focus on HyperCuP, because first, it is the topology we have implemented in our super-peer network, and second, it is very efficient for broadcasts and partitioning which makes it quite suitable as a super-peer topology.

In HyperCuP, the super-peers from a hyper cube (or, more generally, a Cayley graph), thus enabling efficient query broadcasts with guaranteed non-redundancy. Each node issuing a broadcast can be thought of as the root of a specific spanning tree through the P2P network. The topology allows for $\log_2 N$ path length and $\log_2 N$ number of neighbors, where N is the total number of nodes in the network (i.e. the number of

super-peers in our case). Also, a path of $\log_2 N$ length exists between any two super-peers, thus any two distinct schemas can be reached within a short number of hops from each other. See [20] for detailed information about this topology.

Peers connect to the super-peers in a star-like fashion, providing content and content metadata (see Figure 1 for a small HyperCuP topology).

The introduction of super-peers in combination with routing indices reduces the workload of peers significantly by distributing queries only to the appropriate subset of all possible peers (see also [7] who discusses routing indices based on various aggregation strategies of content indices). In our approach, we have introduced two different kinds of routing indices, based on schema information. In the next sections we will discuss these routing indices in detail.

2.1 Routing Super-Peer/Peer Queries and Responses

The first kind of indices needed in super-peers are so-called super-peer/peer routing indices (SP/P-RIs). In these indices each super-peer stores information about metadata usage at each directly connected peer.

On registration the peer provides the super-peer with its metadata information by publishing an advertisement. This advertisement encapsulates a metadata based description of the most significant properties of the peer. As this may involve quite a large amount of metadata, we build upon the schema-based approaches which have successfully been used in the context of mediator-based information systems (e.g. [24]).

To ensure that the indices are always up-to-date, peers notify super-peers when their content changes in ways that trigger an update of the index. If a peer leaves the network, all references to this peer are removed from the indices. In contrast to some other approaches (e.g. Gnutella [10], CAN [19], Tapestry [26]), our indices do not refer to individual content elements but to whole peers (as in CHORD [21]).

At each super-peer, elements used in a query are matched against the SP/P-RIs in order to determine local peers which are able to answer the query (see also [2] for a related approach). A match means that a peer understands and can answer a specific query, but does not guarantee a non-empty answer set. The indices can contain the information about peers (or other super-peers, see 2.2) at different granularities: schema identifiers, schema properties, property value ranges, and individual property values.

To illustrate index usage, we will use the following sample query: *find lectures in German language from the area of software engineering suitable for undergraduates*. In the Semantic Web context this query would probably be formalized using the Dublin Core schema (DC, [4]) for document specific properties (e.g. title, creator, subject) and the Learning Object Metadata schema (LOM, [11]) which provides learning material specific properties, in combination with classification hierarchies (like the ACM Computing Classification System, ACM CCS) in the subject field. In line with RDF conventions citels01, we identify properties by their name and their schema (expressed by a namespace shorthand). "dc:subject" therefore denotes the property "subject" of the DC schema. So, written in a more formal manner, the query becomes:

> *Find any resource where the property* dc:subject *is equal to* ccs:softwareengineering, dc:language *is equal to "de" and* lom:context *is equal to "undergrad".*

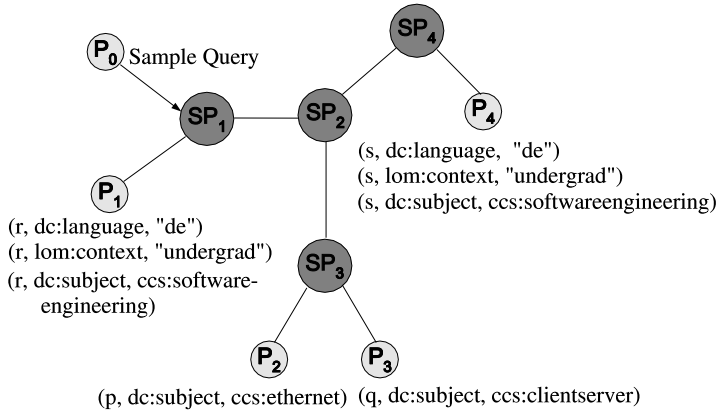

Fig. 2. Routing example network

Table 1 shows the values requested in the query at the different granularities; e.g. the query asks for DC and LOM at the schema level, while it requests a lom:context value of "undergrad" at the property value level, etc.

In order to further clarify things we consider the scenario shown in Figure 2. In this network, various resources are described on different peers, which in turn are attached to super-peers.

Peer P_0 sends the sample query mentioned above to its super-peer SP_1. In our example, this query could be answered by the peers P_1 and P_4, attached to SP_1 and SP_4, respectively. These contain metadata about resources r and s which match the query.

The following paragraphs will explain how the routing indices at the different granularities facilitate routing the query to the right peers.

Schema Index. We assume that different peers will support different schemas and that these schemas can be uniquely identified (e.g. the dc and lom namespaces are uniquely identified by an URI). The routing index contains the schema identifier as well as the peers supporting this schema. Queries are forwarded only to peers which support the schemas used in the query. Super-peer SP_1 will forward the sample query to attached peers which use DC and LOM to annotate resources.

Table 1. Contents of the sample query at different granularities

Granularity	Query	
Schema	dc, lom	
Property	dc:subject, dc:language, lom:context	
Property Value Range	dc:subject	ccs:sw'engineering
Property Value	lom:context dc:language	"undergrad" "de"

Property/Sets of Properties Index. Peers might choose to use only parts of (one or more) schemas, i.e. certain properties, to describe their content. While this is unusual in conventional database systems, it is more often used for data stores using semi-structured data, and very common for RDF-based systems. In this kind of index, super-peers use the properties (uniquely identified by namespace/schema ID plus property name) or sets of properties to describe their peers. Our sample query will be sent to peers using at least dc:subject, dc:language and lom:context (e.g. SP_1 will send the query to P_1, as P_1 contains all of these properties). Sets of properties can be useful to characterize queries (i.e. we might use a "sets-of-properties index" to characterize and route the most common queries).

Property Value Range Index. For properties which contain values from a predefined hierarchical vocabulary we can use an index which specifies taxonomies or part of a taxonomy for properties. This is a common case in Edutella, because in the context of the semantic web quite a few applications use standard vocabularies or ontologies. In our example, peers could be characterized by their possible values in the dc:subject field, and the query would not be forwarded to peers managing "ccs:networks" or "ccs:artificial_intelligence" content (as these sub-hierarchies are disjoint from the ccs:software_engineering sub-hierarchy), and will not be forwarded to peers which use the MeSH vocabulary (because these peers manage medical content).

Note that the subsumption hierarchy in a taxonomy such as ACM CCS can be used to aggregate routing information in order to reduce index size.

Property Value Index. For some properties it may also be advantageous to create value indices to reduce network traffic. This case is identical to a classical database index with the exception that the index entries do not refer to the resource, but the peer providing it. This index contains only properties that are used very often compared to the rest of the data stored at the peers. It would be very interesting to investigate how these indices could be combined with the value mapping approach described in [12].

In the example, this is used to index string valued properties such as dc:language or lom:context.

2.2 Routing among Super-Peers Based on Routing Indices

As with peers, we want to avoid broadcasting queries to all super-peers. To achieve this goal we introduce super-peer/super-peer routing indices to route among the super-peers. These SP/SP indices are essentially extracts and summaries (possibly also approximations) from the local SP/P indices. They contain the same kind of information as SP/P indices, but refer to the (direct) neighbors of a super-peer. Queries are forwarded to super-peer neighbors based on the SP/SP indices, and sent to connected peers based on the SP/P indices.

Table 2 gives a full example of the SP/SP routing index of SP_2 at the different granularities. For example, SP_2 knows at the schema level that all of its neighbors (SP_1, SP_3, SP_4) use the DC namespace, but only SP_1 and SP_4 contain information described in the LOM schema. Thus, the sample query will not be routed to SP_3, as it requires both DC and LOM.

The same applies for the other levels of granularity. A special case is the Property Value Range level; note that ccs:networks is a common super concept of ccs:ethernet and ccs:clientserver in the ACM CCS taxonomy. Making use of the topic hierarchy, the routing index can contain aggregate information like this in order to reduce index size.

Update of SP/SP indices is based on the registration (or update) messages from connected peers. We assume for the moment that a peer can connect to an arbitrary super-peer and define the index update procedure as follows: when a new peer registers with a super-peer, it announces the necessary schema (and possibly content) information to the super-peer. The super-peer matches this information against the entries in its SP/P index. If new elements have to be added in order to include the peer into the SP/P index, the super-peer broadcasts an announcement of the new peer to the super-peer network (according to the HyperCuP protocol, so that it reaches each super-peer exactly once). The other super-peers update their SP/SP indices accordingly. [20] describes the algorithms for joining and leaving of super-peers.

Although such a broadcast is not optimal, it is not too costly either. First, the number of super-peers is much less than the number of all peers. Second, if peers join the super-peer frequently, we can send a summary announcement containing all new elements only in pre-specified intervals instead of sending a separate announcement for each new peer. Third, an announcement is necessary only if the SP/P index changes because of the integration of the new peer. As soon as the super-peer has collected a significant amount of peers (hopefully with similar characteristics, see our discussion on clustering in the next section), the announcements will rather be an exception. We are planning simulations as part of our further work to validate these assumptions quantitatively.

Further work will include simulations to collect data on performance characteristics of the approach.

3 Adaptive Clustering and Mediation in Peer-to-Peer Networks

In the last section we described how queries can be routed using schema-based routing indices. This routing still has the problem that most queries must be broadcast if the peer distribution is arbitrary. In this section we discuss concepts how peers can be

Table 2. SP/SP index of SP_2 at different granularities

Granularity	**Index of SP_2**		
Schema	dc		SP_1, SP_3, SP_4
	lom		SP_1, SP_4
Property	dc:subject		SP_1, SP_3, SP_4
	dc:language		SP_1, SP_4
	lom:context		SP_1, SP_4
Property Value Range	dc:subject	ccs:networks	SP_3
	dc:subject	ccs:software-engineering	SP_1, SP_4
Property Value	lom:context	“undergrad”	SP_1, SP_4
	dc:language	“de”	SP_1, SP_4

smartly clustered in order to avoid unnecessary broadcasting. Furthermore, we propose ideas for mediating information provider peers and consumer peers dynamically by using schema based clustering techniques and for using mediation correspondences to transform between different schemas.

3.1 Rule-Based Clustering of Peers

Obviously, we still have to define what kind of similarity measures we want to use for our partitions. In [20] we have discussed how to partition a HyperCuP-based P2P network based on a topic ontology shared by all peers. Here, we present a new approach called *rule based clustering*. The main idea is to group and register peers in *subject specific clusters (SSC)* via cluster specific rules. Each super-peer represents such a cluster. A typical cluster may group peers with equal properties, e.g. more static properties like specific query and result schemas, specific domain/IP address ranges, or more dynamic properties, like a minimum number of resources at a peer, average answer time or average number of results[2]. Every cluster provides its own *rules*, expressing which peers are allowed to join the cluster and which peers are denied to enter the cluster. We will call a complete set of such rules a *subject specific cluster policy*. Typically the super-peer's administrator will define such a policy[3]. Each rule consists of an event, a constraint and an action. This approach has already been widely used in database systems[23][8]. We distinguish between three different *events*:

Enter A peer likes to join an super-peer.
Leave A peer leaves a super-peer.
Check A super-peer checks the current status of an already connected peer.

An event can be connected to one or many constraints. A typical *constraint* is defined by a property, an operator (=,!=, INCLUDE, EXCLUDE[4]) and a value, e.g.

```
Peer.Advertisement.Property query_schema = "LOM"
```

When checking a constraint, the result can either be "TRUE" or "FALSE". Constraints can be combined using conjunction (AND) and disjunction (OR). As long as a constraint meets our schema, we allow the formulation of arbitrary constraints using arbitrary property sets, since most super-peer administrators will use their own context specific set. If a super-peer receives a peer advertisement consisting of an unknown property, the property is ignored by the super-peer. If a super-peer misses a property in a peer's advertisement while checking the value of a constraint, the result of the constraint is assumed as "FALSE".

[2] Further properties for peers and information sources are discussed in [9][14][22]

[3] We assume that a small number of participants of our P2P network will be competent enough and interested in defining such rules while most of users are only interested in providing and retrieving information from the network. This assumption is supported by the fact that some existing single schema P2P networks, typically only used for file sharing purpose like *Direct Connect* [17] and *E-Donkey*, already use simple administrator based rules for clustering peers successfully.

[4] Further work will include the investigation of more operators, in particular operators which will express fuzzy similarity.

Depending on events and constraints *actions* will be triggered. For instance, the above mentioned event "Enter" will trigger one of the following actions:

Approve (Peer) A peer will be approved to join the super-peer.
Reject (Peer) A peer will be rejected from the super-peer.
Redirect (Peer) A peer will be forwarded to another super peer. Note that this action may occur after approving or rejecting a peer.

In the following example we assume that a super-peer is only interested in peers providing materials by using the LOM or DC schemas and that possible peers have to be member of the domain "cs.tu-berlin.de". The corresponding policy of the super-peer can be expressed by defining one rule[5]:

```
ON (Event) Enter IF (
  ((Peer.Advertisement.Property query_schema="LOM") OR
  (Peer.Advertisement.Property query_schema="DC")
  )
  AND
  (Peer.Advertisement.Property peer_name
        INCLUDE "cs.tu-berlin.de")
)
DO (Action) Approve(Peer)
ELSE (Action) Reject(Peer)
```

Having defined the super-peers clustering policy, the super-peer administrator may be interested in some specific peers which provide relevant information. An approach would be just to wait until peer advertisements occur, denoting peers wishing to join his super-peer. We will also investigate techniques to invite specific peers joining the super peer and to examine automatically advertisements of peers connecting to other super-peers. This includes concepts as broadcasting such invitations periodically to attract matching peers.

Using the proposed clustering approach several problems remain open. One problem is to realize an overall sound clustering. Since rules are specified by local administrators, some peers may not be accepted by neither super-peer at all. This may occur if the peer does not provide a suitable advertisement for a cluster. This may be especially the case when the peer does not meet the minimum requirements of the cluster, e.g. provides only a few results for a query, does not include a specific query schema, does not use a specific ip-address and so on. In our network we explicitly allow a super-peer administrator narrowing its super-peer policy to reject poor information sources. For this reason some peers may not be accepted by any super-peer at all. To avoid 1) dropping to many peers, which maybe are relevant for other super-peers, and 2) help "new peers" to discover a suitable super-peer when entering the network we will investigate in our further research two promising approaches:

[5] The above mentioned examples are described by using a non-existent pseudo language. We will investigate, how existing languages used in the context of the semantic web are feasible to express our semantics.

Meta Data Annotation Current Super-Peer based peer-to-peer applications for music sharing technologies use meta data to describe the policy of each super-peer.[6] Although we did not explicitly specify a model for such annotations yet we believe that such annotations using a dedicated model will help peers identifying a suitable super-peer.

Redirects between Super-Peers Super-Peer administrators may define "links or redirects" between selected super peers. Such links will redirect peers from one super-peer to another if they had been rejected from one super-peer.[7].

3.2 Correspondences-Based Mediation between Different Schemas

Using the criteria of [18] and [3] the super-peer network described in the last sections may be classified as a federated information system (FIS) without a global schema, a so called loosely coupled system. Such systems offer a uniform multi database query language to access data in different sources, but do not have a global schema. Data sources in such systems must be structured and support unrestricted query access. In tightly coupled systems, for instance mediator-based systems [24], users see only one schema and do not have to bother with different sources and their structures. Hence, a tightly coupled FIS inherently offers location, language, and schema transparency. In contrast, loosely coupled systems usually only offer language transparency: a user does not need to learn the query language of each source, but he still has to know their schemas.

In this section we will describe how mediation services may provide schema transparency in our system. We assume that each super-peer will cluster peers for a specific domain using an appropriate clustering policy. Further each peer of the cluster will provide at least one query schema. Unfortunately we cannot assume that every peer will use the same term in its query schema for the same meaning. To resolve this heterogeneity we will introduce in this section local correspondences between different schemas of peers registered at a super-peer. Such correspondences incorporate a local domain mapping logic between schemas of the peers and a global schema of one super-peer. Each super-peer may consist of several correspondences expressing different semantics to correspondences used in other super-peers. Such local correspondences therefore will only resolve heterogeneity within its super-peer.

This new setting might surprise, since in comparison to the setting described in the last sections, where only local schemas are used directly, we now use at least one global schema for each super-peer. We believe that by introducing a local schema in a loosely coupled information system advantages of both strategies, loosely bound information sources and mediation services of tight coupled information systems may be used. Further, clustering peers for a specific domain by query schemas of peers might help to narrow the amount of integration work, in particular formulating correspondences between peer and super-peer schemas for a with a given set of query schemas.

[6] For instance *Direct Connect* provides a list with every super-peer available to join in the network consisting its policy (min of open Slots, min available GB amount of data, allowed IP address range, name of the peer-mostly includes information about its shared music styles...).

[7] We are aware of only one super-peer network, Direct Connect, which uses this approach to redirect between different hubs.

In our further research we will investigate how such a tight integration of clustering and mediation concepts may be used in our peer-to-peer infrastructure providing richer queries.

For our mediation service we assume that every peer will provide information about its query schema in an advertisement. Typically a super-peer will collect several advertisements related to its peers.[8] If a super receives a query from a consumer it tries to identify relevant advertisements matching the schema of the query. We distinguish between the following three cases:

1. A query exactly matches the advertisement of one potential peer.
2. A query exactly matches advertisement of many peers, all using the same schema.
3. A query could be resolved combining results from many peers using different schemas.

Case one and two may occur when a super-peer's clustering policy forces its peers to use only one schema. Since we allow heterogeneous schemas in our super-peer network we are mainly interested in case three which includes case one and two already. This implies we have to investigate methods to transform schemas between different peers, so different query schemas can be integrated with each other.

Consider the example where a super-peer has defined *subject specific clustering rules (SSCRs)* (see also section3.1) and now accepts only peers using either LOM or DC schema. When receiving queries consisting of LOM and DC specific attributes the super-peer has to translate between the attributes of LOM and DC. To resolve these heterogeneities, we investigated concepts for transformation rules between different schemas, so called *correspondences*, already used in mediator-based information systems (MBIS) [24]. We identified *Query Correspondence Assertions (QCA)*[13] and *model correspondences (MOCA)*[6] as a flexible mechanisms to express such correspondences between heterogeneous schemas. With QCAs, a human administrator defines the intentional equivalence of two views, where one is defined as a query against the mediator schema and the other is defined as a query against one source schema. In contrast to MBIS where correspondences are used as rules to translate between global and local schemas, in super-peer networks typically translations exist between different local schemas only. Such MBIS-based correspondences can also be used as rules to describe such translations. We will adapt the existing concepts of MOCAs and QCAs as a possible way to define schema correspondences in peer-to-peer networks and describe their use in our network by an example.

In the following example the super-peer administrator defines a query schema *lectures(lecture:identifier, lecture:language, lecture:subject, lecture:educationalcontext)* which returns documents identified by its URL. First we define correspondences between attributes of the peer schemas and the corresponding attributes of the *lectures* schema[9]:

1. lectures:Identifier = dc:identifier
   lectures:language=dc:lang
   lectures:subject=dc:subject

[8] Depending on the super-peer's policy, peer advertisements may contain many different schemas.

[9] The correspondences are based on existing LOM-DC mappings, see *http://kmr.nada.kth.se/el/ims/md-lomrdf.html*.

2. lectures:Identifier = lom:general.identifier
   lectures:language=lom:general.language
   lectures:context=lom:educational.context

Using the above mentioned correspondences we now create views on the peer specific schemas:

1. lecturesViewDC(lectures:Identifier,lectures:language,lectures:subject)
   ← DC(dc:Identifier, dc:lang, dc:subject)
2. lecturesViewLOM(lectures:Identifier, lectures:language, lectures:context)
   ← LOM(lom:general.identifier,lom:general.language, lom:educational.context)

Then we describe, which attributes of the super-peers lectures schema could be answered by the local peer schemas:

1. lectures(lectures:identifier,lectures:language,lectures:subject,-)
   ← lecturesViewDC(lectures:Identifier,lectures:Language,lectures:subject)
2. lectures(lecture:identifier,lecture:language,-, lecture:context)
   ← (lectures:Identifier,lectures:Language,lectures:context)

Combining all correspondences results in two main schema correspondences bridging the heterogeneity between the peers P1 and P2:

Peer1:Correspondence1 lectures(lectures:identifier,lectures:language,-,lectures:edu...context)
← v(lectures:Identifier,lectures:language,lectures:context)
← LOM(lom:general.identifier,lom:general.language,lom:educational.context)
Peer2:Correspondence2 lectures(lectures:identifier,lectures:language,lectures:subject,-)
← v(lectures:Identifier,lectures:language,lectures:subject)
← DC(dc:identifier,dc:subject,dc:lang)

A super-peer stores relations between correspondences and peers in his indices. When a super-peer receives a query *lecture (lecture:identifier, lecture:language, lecture:subject, lecture:educationalcontext)* the super-peer identifies P1:Correspondence1 and P2:Correspondence2 as a combination of relevant correspondences that are semantically included in the user query and will probably compute correct results. The query is forwarded to the relating information provider peers Peer 1 and Peer 2, then the results have to be collected and combined by the super-peer. Integrating these concepts in our super-peer network allows us to build up subject and context specific super-peers in our network. Consider the example where a super-peer administrator is interested in clustering e-learning content providers (see Figure 3). By defining the super-peer's policy using rule based clustering he allows peers connecting to his super-peer only when they provide LOM or DC schema metadata. Next the administrator defines which complex query schemas his super-peer supports and defines the correspondences between these schemas. Finally he invites a first set of relevant information provider peers to join his super-peer. Now the super-peer is ready to receive queries related to learning materials. Other provider peers may join the super-peer later and increase the content mediated through this super-peer.

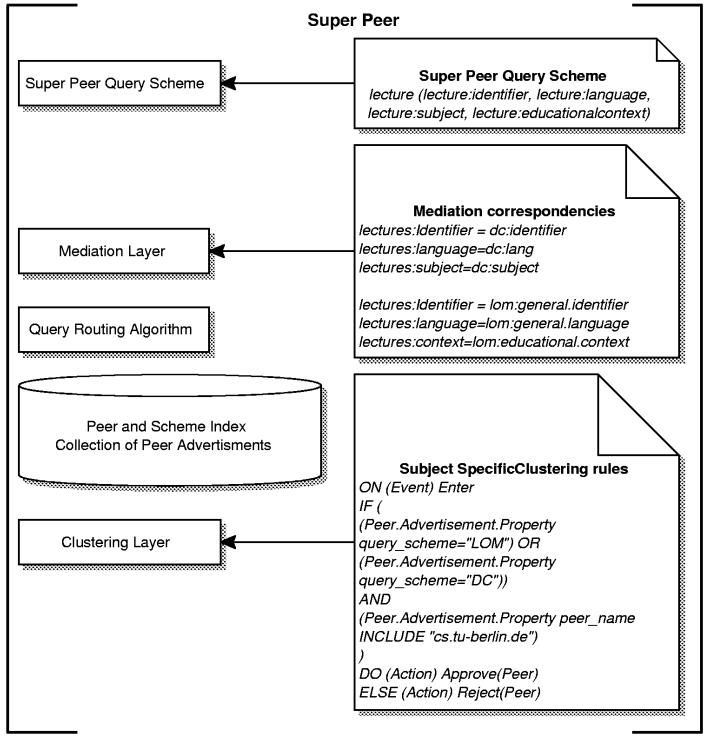

Fig. 3. Context specific Super Peer for E-Learning Materials

4 Conclusion

Schema-based P2P networks have a number of important advantages over previous simpler P2P networks. Peers in these networks provide and use explicit (possibly heterogeneous) schema descriptions of their content, thus allowing us to build up an infrastructure ideally suited to connect heterogeneous information providers.

We proposed a super-peer topology as a suitable topology for these schema-based P2P networks and discussed how the schema information can be used for routing and clustering in such a network. In our approach super-peer indices exploit the RDF ability to uniquely identify schemas, schema attributes and ontologies, and are used for routing between super-peers and peers as well as within the super-peer backbone network.

Combining rule based clustering and correspondences between different schemas is used to adaptively collect and filter heterogeneous information sources and to integrate them in a context specific way mediated through super-peers. This approach combines the dynamic self organizing behavior of peer-to-peer networks with existing information integration concepts of mediator-based information systems. In comparison to traditional mediator-based information systems, information consumers and information providers can connect dynamically and schema usage can be extended dynamically.

Rule based clustering in large heterogeneous super-peer networks can cluster peers efficiently to avoid broadcasting and flooding the network with queries. Subject specific clustering techniques can create peer based ontologies of information consumers and information providers for a specific context.

References

1. K. Aberer, P. Cudré-Mauroux, and M. Hauswirth. The chatty web: Emergent semantics through gossiping. In *Proceedings of the Twelfth International World Wide Web Conference (WWW2003)*, Budapest, Hungary, May 2003.
2. K. Aberer and M. Hauswirth. Semantic gossiping. In *Database and Information Systems Research for Semantic Web and Enterprises, Invitational Workshop*, University of Georgia, Amicalola Falls and State Park, Georgia, April 2002.
3. A.Sheth and J. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22(3):183–236, 1990.
4. D. Beckett, E. Miller, and D. Brickley. Expressing simple dublin core in RDF/XML. Technical report, Dublin Core Metadata Initiative, 2002. `http://dublincore.org/documents/2002/07/31/dcmes-xml/`.
5. P. A. Bernstein, F. Giunchiglia, A. Kementsietsidis, J. Mylopoulos, L. Serafini, and I. Zaihrayeu. Data management for peer-to-peer computing: A vision. In *Proceedings of the Fifth International Workshop on the Web and Databases*, Madison, Wisconsin, June 2002.
6. S. Busse. *Model Correspondences in Continuous Engineering of MBIS - doctorial thesis*. Logos Verlag, September 2002.
7. A. Crespo and H. Garcia-Molina. Routing indices for peer-to-peer systems. In *Proceedings International Conference on Distributed Computing Systems*, July 2002.
8. U. Dayal, E. N. Hanson, and J. Widom. Active database systems. In *Modern Database Systems*, pages 434–456. ACM SIGMOD International Conference on Management of Data, 1995.
9. H. Garcia-Molina and B. Yang. Efficient search in peer-to-peer networks. In *Proceedings of ICDCS*, 2002.
10. Gnutella.
11. IEEE P1484.12 Learning Object Metadata Working Group. Draft standard for learning object metadata. Technical report, IEEE Learning Technology Standards Committee (LTSC), 2002. `http://ltsc.ieee.org/doc/wg12/LOM_1484_12_1_v1_Final_Draft.pdf`.
12. A. Kementsietsidis, M. Arenas, and R. J. Miller. Mapping data in peer-to-peer systems: Semantics and algorithmic issues. In *Proceedings of the ACM SIGMOD International Conference on Management of Data)*, June 2003.
13. U. Leser. *Query Planning in Mediator Based Information Systems – doctorial thesis*. TU Berlin, June 2000.
14. F. Naumann. *Quality-Driven Query Answering for Integrated Information Systems - doctorial thesis*. Springer Verlag, lecture notes in computer science, 2261 edition, July 2002.
15. W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmér, and T. Risch. EDUTELLA: a P2P Networking Infrastructure based on RDF. In *WWW 11 Conference Proceedings*, Hawaii, USA, May 2002.
16. W. Nejdl, M. Wolpers, W. Siberski, A. Löser, I. Bruckhorst, M. Schlosser, and C. Schmitz. Super-Peer-Based Routing and Clustering Strategies for RDF-Based Peer-To-Peer Networks. In *Proceedings of the Twelfth International World Wide Web Conference (WWW2003)*, Budapest, Hungary, May 2003.
17. Neo-Modus. Direct Connect Homepage. http://www.neo-modus.com/.

18. M. Oezsu and P. Valduriez. *Principles of distributed database systems*. Prentice Hall, 2nd edition edition, 1999.
19. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content addressable network. In *Proceedings of the 2001 Conference on applications, technologies, architectures, and protocols for computer communications*. ACM Press New York, NY, USA, 2001.
20. M. Schlosser, M. Sintek, S. Decker, and W. Nejdl. HyperCuP—Hypercubes, Ontologies and Efficient Search on P2P Networks. In *International Workshop on Agents and Peer-to-Peer Computing*, Bologna, Italy, July 2002.
21. I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the 2001 Conference on applications, technologies, architectures, and protocols for computer communications*. ACM Press New York, NY, USA, 2001.
22. D. M. Strong, Y. W. Lee, and R. Y. Wang. Data quality in context. *Communications of the ACM*, 40(5):103–110, 1997.
23. J. Widom and U. Dayal. *A Guide To Active Databases*. Morgan-Kaufmann, 1993.
24. G. Wiederhold. Mediators in the architecture of future information systems. *IEEE Computer*, 25(3):38–49, 1992.
25. B. Yang and H. Garcia-Molina. Designing a super-peer network. `http://dbpubs.stanford.edu:8090/pub/2002-13`, 2002.
26. B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical Report UCB/CSD-01-1141, UC Berkeley, EECS, 2001.