

# Semantic Overlay Clusters within Super-Peer Networks

Alexander Löser<sup>1</sup>, Felix Naumann<sup>2</sup>, Wolf Siberski<sup>3</sup>, Wolfgang Nejdl<sup>3</sup>, Uwe Thaden<sup>3</sup>

<sup>1</sup> CIS, Technische Universität Berlin, 10587 Berlin, Germany  
aloeser@cs.tu-berlin.de

<sup>2</sup> Computer Sciences, Humboldt University Berlin, 10099 Berlin, Germany  
naumann@informatik.hu-berlin.de

<sup>3</sup> Learning Lab Lower Saxony, 30167 Hannover  
siberski,nejdl,thaden@learninglab.de

**Abstract.** When joining information provider peers to a peer-to-peer network, an arbitrary distribution is sub-optimal. In fact, clustering peers by their characteristics, enhances search and integration significantly. Currently super-peer networks, such as the Edutella network, provide no sophisticated means for such a "semantic clustering" of peers. We introduce the concept of semantic overlay clusters (SOC) for super-peer networks enabling a controlled distribution of peers to clusters. In contrast to the recently announced semantic overlay network approach designed for flat, pure peer-to-peer topologies and for limited meta data sets, such as simple filenames, we allow a clustering of complex heterogeneous schemes known from relational databases and use advantages of super-peer networks, such as efficient search and broadcast of messages. Our approach is based on predefined policies defined by human experts. Based on such policies a fully decentralized broadcast- and matching approach distributes the peers automatically to super-peers. Thus we are able to automatize the integration of information sources in super-peer networks and reduce flooding of the network with messages.

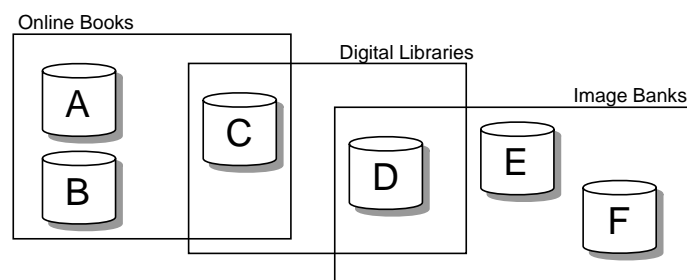
## 1 Introduction

Current peer-to-peer (P2P) networks support only limited meta data sets such as simple filenames. Recently a new class of peer-to-peer networks, so called schema based peer-to-peer networks have emerged (see [2, 11, 4, 20, 15]), combining approaches from peer-to-peer research as well as from the database and semantic web research areas. Such networks build upon peers that use explicit schemas to describe their content. The meta data describing peers is based on heterogeneous schemata. They allow the aggregation and integration of data from autonomous, distributed data sources. However current schema-based peer-to-peer networks have still the following shortcomings:

- Schema based P2P networks that broadcast all queries to all peers don't scale. Intelligent routing- and network organization strategies are essential in such networks so queries are only routed to a *semantically chosen subset of peers* able to answer parts or whole queries. First approaches to enhance routing efficiency in a clustered network have already been proposed by [21] and [23].

- For most domains usually only a small but well-defined set of meta data standards exists. Peers provide information using such standards. For bridging the heterogeneity between different meta data schemes within the domain, mappings have to be provided. *Clustering peers by their schemes* enables the efficient reuse of such existing mappings within a particular domain.

Both issues, forwarding complex queries to selected peers and integration of small groups of schemas for a particular context benefit either from a search-driven or integration-driven clustering of the network in logically portions. Figure 1 shows peers clustered by their characteristics. There are many challenges building such semantic overlay clus-



**Fig. 1.** Semantic Overlay Clusters

ters: What are suitable models describing the nodes and clusters? How can such models be matched in a distributed environment? What is a suitable topology? This paper addresses those questions by introducing deeper to semantic overlay clusters (section 3.2), presents a model for information provider peers (section 4), describes clustering policies for describing the demand on peers for an particular cluster (section 5) and shows matching and broadcasting approaches (section 6).

## 2 Related Work

In previous papers [15][20][19], we have described an RDF-based P2P infrastructure called Edutella (see <http://edutella.jxta.org> for the source code). It aims at providing access to distributed collections of digital resources through a P2P network. The idea of placing data nodes together, so queries can be efficiently routed and a semantic integration of the nodes is more automatized, has been discussed in many research projects. In the field of federated databases the tightly coupled mediator-wrapper architecture [26] was proposed by Wiederhold, enabling a static integration of domain-specific information sources. Matchmaking Infrastructures, such as InfoSleuth [14] or OBSERVER[16], match information provider to information consumers in a centralized way using description logics. In the Artificial Intelligence field the conceptual clustering problem has been widely studied in inductive learning systems, such as in COBWEB[6] and LABYRINTH [25]. Other approaches for routing queries directly to existing clusters

are proposed by [21]. However, most systems assume that documents are part of a controlled collection located at a central database and allow only a centralized matching. Recently semantic overlay networks for peer-to-peer systems [23] allow overlays for placing data nodes semantically together. However they allow only the use of limited meta data schemes, such as simple filenames, and are designed for pure peer-to-peer networks, without using advantages of super-peer networks.

### 3 Clustering in Super-Peer based Networks

In this section we show the use of super-peer networks for a semantic clustering of information provider. After a short introduction to super-peer networks we present the concept of semantic overlay clusters and an extension to an existing super-peer infrastructure enabling such clusters.

#### 3.1 Super-Peer Networks

Recently a new wave of peer-to-peer systems is advancing an architecture of centralized topology embedded in decentralized systems; such a topology forms a super-peer network. Super-peer networks occupy the middle-ground between centralized and entirely symmetric peer-to-peer networks. They introduce hierarchy into the network in the form of super-peer nodes, peers which have extra capabilities and duties in the network (see [9]). A super-peer is a node that acts as a centralized server to a subset of clients, e.g. information provider and information consumer. Clients submit queries to their super-peer node and receive results from it, as in a hybrid system. However, super-peers are also connected to each other as peers in a pure system are (see also figure 2), routing messages over this overlay network, and submitting and answering queries on behalf of their clients and themselves. Examples of super-peer networks are JXTA[10], Edutella[19] or Morpheus. Because a super-peer network combines elements of both

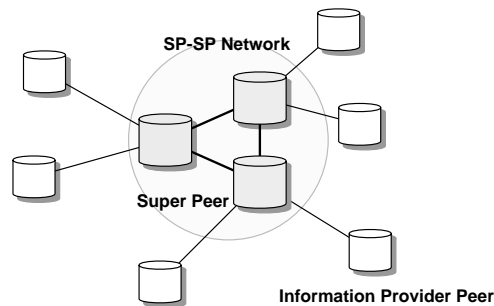


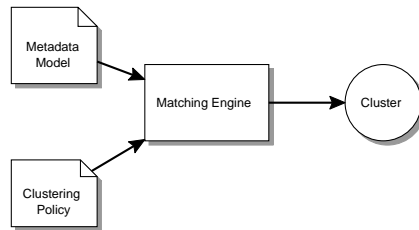
Fig. 2. Super-Peer Network

pure and hybrid systems, it has the potential to combine the efficiency of a centralized search with the autonomy, load balancing[27], robustness to attacks and at least semantic interoperability [2] provided by distributed search.

### 3.2 Semantic Overlay Clustering

In this section we introduce the concept of semantic overlay clusters (SOC). Existing super-peer networks do not provide capabilities for enabling the definition and construction of SOC's yet. However some existing super-peer networks already provide clusters based on the physical network topology, such as JXTA with its group model or Edutella (see [20],[15]). In a super-peer networks a set of clients together with their super-peer forms a cluster. Intra cluster data communication takes place via direct peer to peer links between the clients, inter cluster communication takes place via links between super-peers. So far all the above described methods do not describe the structure of the clusters semantically. For enabling SOC as logical layers above the physical network topology we need a clustering method suitable to match semantically information provider peers to super-peer based clusters. Similar to the definition for semantic overlay networks by [23] we assume existing information provider peers and existing super peers as nodes in a physical network. Both can exchange messages within the network. A semantic overlay cluster ( $SOC_l$ ) is defined as a link structure within a physical network ( $N$ ) given a set of links from information provider ( $p$ ) to a particular super-peer ( $s$ ): ( $SOC_l = p_i, s_j \in N | \existsalink(p_i, s_j, l)$ ). In addition we assume that each  $SOC_l$  supports at least two functions:  $Join(p_i, l)$ , where links ( $p_i, s_j, l$ ) between a super-peer and a information provider peer are created and  $Leave(p_i, l)$  where they are dropped.

We focus our work on the realization of the *Join* function. Requests for a join are made by issuing a meta data based model  $m_i$  of a particular  $p_i$  to the network. We assume that every information provider provides such a model. Furthermore each cluster is related to one super-peer  $s_j$  and expresses explicitly its demand for information provider peers by a so called clustering policy  $c_j$ . We model a match between a clustering policy  $c_j$  and an the model of an information provider  $m_i$  as a function  $Match(m_i, c_j)$  that returns 1 if there is a match and 0 otherwise (see also figure 3). The total number of matches  $T$  for an particular model  $m$  is the number of matches over all clustering policies:  $T(m, c_j) = \sum_j Match(m, c_j)$ . Matches can either be exhaustive, partial, fuzzy or ontology-based.



**Fig. 3.** Metadata Modell-based Clustering Approach

Now we look closer at the components enabling SOC's in a super-peer network:

- **information provider model** The models  $m_i$  contain a semantic rich description of the underlying peer, including (among others) information about the query and export schema of the peer, quality aspects and classification aspects. Furthermore they should be extensible by application specific annotations. We need to define a schema for these models and also need to ensure that they can be handled at the super-peers.
- **clustering policies** Policies  $c_j$  describe constraints on information provider peers for each cluster. We use policies to select automatically particular sources from all available information sources, taking into account the underlying model of the information source. Since policies are defined by an human expert, they have to be formalized in some way, so algorithms can match suitable information provider automatically.
- **matching engines** Information provider model and clustering policies are matched against each other by a matching function. If a match occurs, a peer joins a super-peer. Matching is detected by a matching engine which implements the matching function  $Match_j()$ . Matches can either be exhaustive, partial, fuzzy or ontology-based. We do not assume a common matching engine. Rather, each super-peer may select its own matching concepts and local engine implementation, depending on its needs.
- **model distribution engine** Since each super-peer owns a separate implementation of a "personal" matching engine and its specific super-peer dependent clustering policy, models of information provider peers willing to join one or more super-peers are distributed to all super-peers in the in super-peer network. This is done by a broadcast.

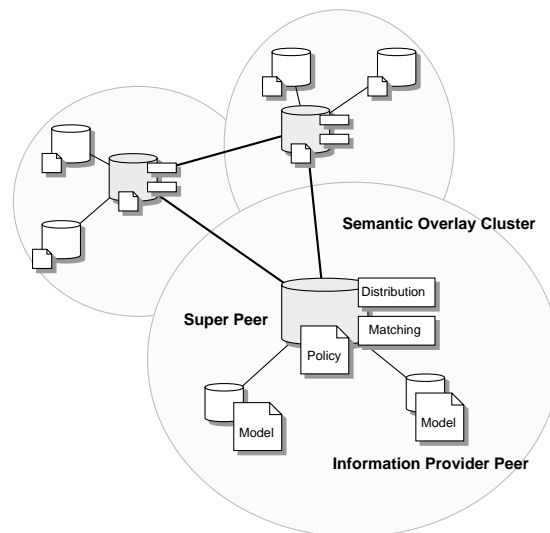


Fig. 4. Super-Peer Network with Clustering Policy and Information Provider Model

Figure 4 illustrates the extension of the "traditional" super-peer architecture. Each super-peer represents a separate semantic overlay cluster. Information provider peers are extended by an information provider model. Super-peers, typically computer with loads of memory and processing power, are extended with the clustering policy, matching and distribution concepts, Furthermore, in this figure not shown, information provider peers may join two or more super-peers.

In the following sections we will describe the elements of our approach in detail. In section 4 we present our information provider model. Section 5 discusses the description of clustering policies and their relation to the information provider model. Section 6 shows how the matching process works for new peers joining the network.

#### 4 The Information Provider Model

The metadata model presented in this section provides an annotation schema designed to support the definition of semantic overlay clusters by local domain experts within the Edutella Network. This model shows a set of attributes for a particular infrastructure. In a semantic overlay cluster environment the model is used for the identification of relevant information provider peers. It consists of 15 attributes, which are either extracted from the information provider peer automatically at runtime (Peer ID, Peer IP, Peer Domain, Completeness, Accuracy, Response Time, Amount of Data) or are manually defined by local domain experts (Peer Schema, Peer Name, Peer Description, Global Classification URI and Taxon Path).The model (see figure 5) consists of five RDF Classes containing several annotations, e.g. annotations for information provider peers such as schema based annotations used in mediator-based information systems, annotations for

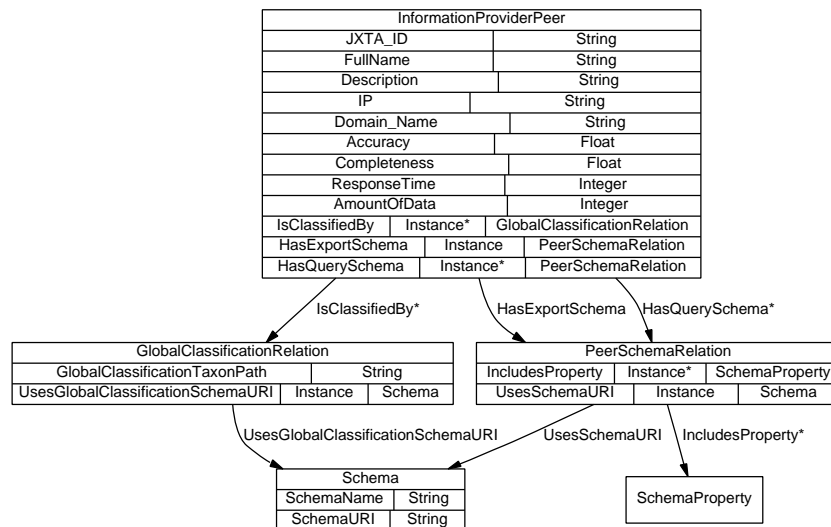


Fig. 5. Edutella Information Provider Peer Metadata Model (extract)

information quality used in the context of federated information systems, peer-to-peer specific annotations and annotations for classifying peers according to existing taxonomies. The complete model can be taken from <http://nutria.cs.tu-berlin.de/edutella/>. In the following subsections we show the model details.

Since there is no ideal model describing arbitrary information sources, our model should be viewed as a core of relevant attributes; it may be completed by attributes from other models, as with any other RDFS based schema. Other systems do also use peer models to improve the peer-to-peer network characteristics. [9] uses a metric model for improving search in peer-to-peer networks, including annotations such as average aggregate bandwidth, average aggregate processing cost, number of results, satisfaction of the query and time to satisfaction. Semantic characteristics are not taken into account. [13] proposes a model for encoding semantic information as content categorization, security information, visibility of resources at a peer and caching of resources. This model is similar to ours, but it isn't used for clustering and it doesn't contain schema and quality information. The model used in [23] consists of one or several content classifications which are used to form semantic overlay networks, also to avoid searching on nodes that have unrelated content.

#### 4.1 Annotations for a Peer Classification

Classification annotations include mainly information about the peer location, human readable description and its classification within existing taxonomies. We distinguish between the following attributes:

**Peer ID** This ID represents a unique identifier of the peer within the network. Since we use the JXTA platform[10] as underlying P2P infrastructure we use the JXTA ID URN.

**Peer Description** Human readable label, describing the purpose of the peer.

**Peer IP** The underlying IP of the peer.

**Peer Domain** The full qualified domain name, e.g. *nutria.cs.tu-berlin.de*

**Peer Name** This label contains information human readable information about the peer, e.g. *E-Learn Repository TU Berlin*

**Global Classification Scheme URI** A major problem when classifying a information provider peer is to find a suitable global classification scheme or taxonomy. In the world wide web the classification of web sites has been widely adopted. Examples are Yahoo and DMOZ. This label contains the URL of any recognized "official" taxonomy or any user-defined taxonomy, e.g. *http://www.dmoz.org*

**Global Classification Scheme TaxonPath** This label represents an entry in a classification as a path from a more general to more specific entry in a classification, e.g. *Programming/Methodologies/Modeling\_Languages/UML/Education/*

#### 4.2 Annotations to Schema Information

Such annotations include schema information such as schemas or attributes used, as well as possibly conventional indexes on attribute values. We build upon the schema-based approaches successfully used in the context of mediator-based information systems [26]. Elements used in a query are matched against the schema information for a

particular information provider peer in order to determine if the information provider peer is able to answer the query, see also [3] and [20] 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. Schema information contain information about query capabilities for a particular peer at different granularities: schema identifiers, schema properties, property value ranges, and individual property values (we already resented concepts in [15] and [20]).

**Schema Index** We assume that different peers support different schemas and that these schemas can be uniquely identified. 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. An example are the `dc` and `lom` namespaces, they are uniquely identified by an URI.

**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. Examples are `dc:subject`, `dc:language` and `lom:context`. In our model we used the semantics of <http://www.w3.org/1999/02/22-rdf-syntax-ns/Property>.

**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. Examples are `dc:subject=ccs:networks`.

**Property Value Index** For some properties it may also be advantageous to create value indexes 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. Examples are `lom:context=undergraduate` or `dc:language=DE`.

### 4.3 Annotations for Information Quality

In recent times both researchers and practitioners have recognized that reasoning about information quality has become one of the most important tasks when integrating information from autonomous information sources, such as information provider peers [18]. In the following paragraphs, we list information quality criteria that are relevant for the classification of read-only type information sources, like peers. Additionally we provide a short description of how we assess the scores for these criteria.

**Completeness** For an information provider peer, completeness is a measure for the “size” of the underlying data source. The size of an information provider peer is measured as the absolute number of available resources. This number is usually provided by the information provider themselves as a form of advertisement. Information provider peers with a higher completeness are of higher quality to users, because the probability to find a suitable resource is higher.

**Accuracy** is the quotient of the number of correct values in a source and the overall number of values in the source. A value is an instance of an attribute. For our context accuracy is the percentage of data without *data errors*, such as non-unique keys or out of range values. Mohan et al. give a list of possible data errors [17].

Accuracy has been subject of several research projects [12, 7]. The impact of data errors on data mining methods and data warehouses gives rise to data cleansing methods. The methods identify and eliminate a variety of data errors. The identification techniques can be used to count errors and thus to assess data quality.

**Response Time** measures the average delay in milliseconds between submission of a request and reception of the complete response from the information provider peer. The score for this criterion depends on unpredictable factors, such as network traffic, server workload etc. Also, the technical equipment of the information server plays a role as well. Response time can be automatically assessed through *query calibration*; statistics about average response time under different circumstances and times are gathered. They can be updated with each call to the information provider peer and are thus quite accurate.

**Amount of Data** is the size of the query result, measured in bytes. In contrast to the completeness criterion, amount of data is considered a cost factor; a higher amount of data means more storage and bandwidth needs. Just like response time, amount of data can be assess through the gathering and updating of statistics during actual calls to the information provider peer.

Of course, the list above is only a subjective choice of quality dimensions. Different application domains might need other criteria. For instance, information provider peers based on a fee should include *price* as a cost dimension. For processing-type information provider peers, which are not covered here, different information quality criteria are of importance. Examples of such criteria include *security*, *availability*, and *reliability*.

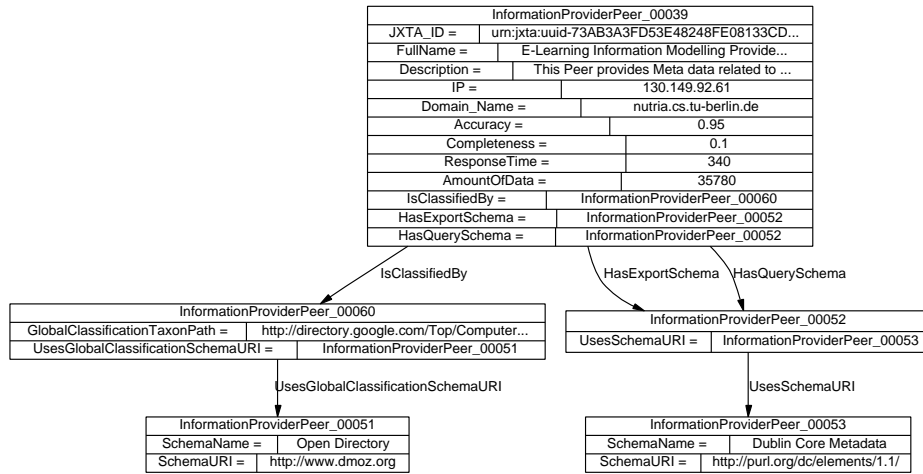
## 5 Clustering Policies

Clustering policies express the demand of information provider peers for a particular application domain. They are defined manually by local domain experts. In super-peer networks each super-peer represents a cluster of domain specific information provider peers and is related to exact one clustering policy. Every cluster policy consists of *rules*, expressing which information provider peers are allowed to join the cluster and which services are denied to enter the cluster. Each rule consists of an event, a constraint and an action. Table 1 shows five rules we identified so far. An event can be connected to one ore many constraints. A typical *constraint* is defined by a property, an operator <sup>4)</sup> and a value, e.g. `Peer.Advertisement.Property accuracy > .95`. When checking a constraint, the value of the check can be either "TRUE" or "FALSE". In the following example we assume, that a super-peer is only interested in information provider peers providing URLs and metadata of materials related to "UML Education" by using the Dublin Core scheme as export schema, having an accuracy of more then 95 per cent and are classified according the Open Directory (see also figure 6 as an example

<sup>4</sup> E.g.: =, !=, <, >, INCLUDE, EXCLUDE, SIMILAR-TO, PART-OF-ONTOLOGY,...

No	Event	Constraint	Action	Explanation
1	Enter	True	Approve	a new service is accepted at the cluster
2	Enter	False	Reject	a new service is not accepted and is rejected
3	Leave	-	DeleteEntry	an registered service leaves the cluster
4	Check	True	-	an registered service is re-accepted
5	Check	False	Reject	an registered service is rejected from the cluster

**Table 1.** Possible rules within a clustering policy



**Fig. 6.** Edutella Information Provider Peer Example

for such an information provider peer). The corresponding policy of the super-peer can be expressed by defining one rule.<sup>5</sup>:

```

ON (Event) Enter

IF (
  (Peer.Advertisement.Property
  HasExportScheme.SchemaURI="http://purl.org/dc/elements/1.1/" )

  AND (Peer.Advertisement.Property accuracy > .95 )

  AND (Peer.Advertisement.Property IsClassifiedBy.URI
  "http://www.dmoz.org" )

  AND (Peer.Advertisement.Property
  IsClassifiedBy.GlobalClassificationTaxonPath INCLUDES
  "Programming/Methodologies/Modeling_Languages/UML/Education/" )

```

<sup>5</sup> The above mentioned examples are described by using a non existent pseudo language, similar to Java.

)

DO (Action) Approve(service)

Constraints can be combined conjunctive (AND) and disjunctive (OR). As long as a constraint meet our scheme, we allow the formulation of arbitrary constraints using arbitrary property sets, since most super-peer administrators use their own context specific set. If a super-peer receives an service advertisement consisting an unknown property, the property is ignored by the super-peer. If a super-peer misses a property in a service advertisement while checking the value of a constraint, the value of the constraint is assumed as "FALSE".

## 6 Matching and Distributing Metadata Models

In the previous sections we presented concepts extending "classical" super-peer networks. In such a network each super-peer consists of its own clustering policy. Furthermore we allow at each super-peer a local matching engine supporting different kinds of matches. Such local matching engines may implemented by the super-peer administrator according to the domain and context of the super-peer. Since both, clustering policies and matching engines, are distributed over the hole super-peer network the matching process between clustering policies and information provider peers models is two folded:

- Broadcast of the information provider peer model within the whole super-peer network to all super-peers
- Matching of the information provider peer model with each local super-peer specific clustering policy according to the local implemented matching engine

In this section we show approaches for distributing information provider peer models within the super-peer network and show possible matching strategies matching models and clustering policies.

### 6.1 Distribution

For joining the network an information provider peer chooses an arbitrary super-peer in the network and forwards its model to the super-peer. The super-peer executes two operations, it first matches the model against its clustering policy and allows or denies the join of the peer to its cluster, second it broadcasts the model to all other super-peers in the super-peer network (see figure 7). A broadcast of a model should include a forwarding of the model to all super-peers in the network, every super-peer should only receive the model once. This can be achieved by computing the minimal spanning tree (MST) over the super-peer network from the initiating super-peer. Building a MST is a well-studied problem (see [8]). In the peer-to-peer community this problem has been addressed by many search and broadcast algorithms. Since in super-peer networks the "inner" network is a pure peer-to-peer network, we use an existing algorithm.

There are only a few algorithms which broadcast messages to all peers with a minimum overhead for a very large number of nodes. DHT-based Algorithms, like CAN

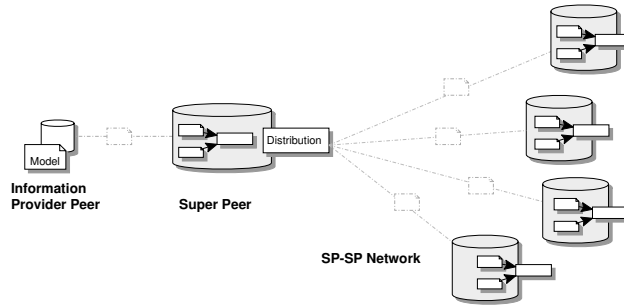


Fig. 7. Matching and distribution of models in the Super-Peer Network

and CHORD, are developed for simple models of resources, e.g. key value pairs for file sharing, and allow therefore not a broadcast of complex models of information provider peers. Schlosser et.al presented the HyperCuP [22] a highly scalable topology which enables efficient broadcast and search algorithms without any message overhead at all during broadcast, logarithmic network diameter, and resiliency in case of node failures. It is guaranteed that exactly  $N-1$  messages are required to reach all nodes in a topology. Furthermore, the last nodes are reached after  $\log_b N$  forwarding steps. Any node can be the origin of a broadcast in the network, satisfying a crucial requirement. The algorithm works as follows: A node invoking a broadcast sends the broadcast message to all its neighbors, tagging it with the edge label on which the message was sent. Nodes receiving the message restrict the forwarding of the message to those links tagged with higher edge labels. Other approaches we identified so far are Bayeux, Zhunag et.al.[28], and P-Grid, Aberer [1] <sup>6</sup>.

## 6.2 Matching

Matchmaking concepts between models and clustering policies depend from domain and goal of the super-peer. This includes matching operators at attribute level and matching algorithms behind operators. We can not assume that in the near future a "One size fits all" approach will be available. Therefore each super-peer uses its own matching engine. A matching engine matches an information provider peer model against the local super-peer policy. Its interface should include the method  $float r = match(profile p, model m)$ , with  $0 \leq r \leq 1$ . We distinguish so far between four concepts of matchings:

- **Exact** In this case an information provider peer only joins a super-peer when its model matches exact with the clustering policy,  $r$  can be either 0 or 1.
- **Partial** The information provider peer may also join the super-peer if only some attributes of the model match with the clustering policy. The result of the match  $r$  is calculated as  $r = \frac{NumberOfMatchingConstraints}{NumberOfAllConstraints}$ . Matching engines for exact and partial matches may be implemented using an RDF-Query language for

<sup>6</sup> Further algorithm may exist.

the RDF-based information provider peer model. Such matchings concepts could be used for the attributes *IP*, *ResponseTime*, *Accuracy*, *AmountOfData* and *Completeness*. Matching operators for such matching concepts are for instance =, !=, <, >, *INCLUDE* and *EXCLUDE*.

- **Similar** For same attributes of the model, such as *Description* and *Fullname*, an exact or partial match is sometimes not be successful, e.g. if a description contains the phrase "Database materials" but the policy looks for "data base materials". Both literals express the same thing, but use different syntax. A match for such attributes occurs if these attributes are syntactically/verbatim similar to the constraints of the policy, expressed by the operator *SIMILAR – TO*. This field has been widely studies in the past. Most search engines such as Google and SMART [5] equate text similarity with content similarity and use keywords and verbatim phrases to identify similar/relevant documents. The result *r* of the match expresses the similarity, an *r* near to 0 expresses a low overlapping, an *r* near 1 a high overlapping between attributes of the model and the clustering policy.
- **Ontology** This more sophisticated approach includes the collection of attributes which are part of an ontology. Consider the case, a super-peer might be interested in clustering restaurant providers for a specific geographic region. First it has to decide, whether an information provider peer offers materials for this area and uses words as Café, Bar, Tourism and so far, second has to relate the restaurant to an specific geographic area. Existing ontologies such as ChefMoZ could be used to define concepts and relations. An attribute of an information provider peer should consist of a relation to the ontology, at least a *IsPartOf* relation, e.g. by using the operator *PART – OF – ONTOLOGY*. Calculating the result is difficult, since different relations between different concepts result in different measures of similarity. First approaches for an ontology-based matchmaking have been shown recently by [24].

Since the information provider peer descriptions are based on RDF annotations, and clustering policies could be understood as queries over an RDF graph an straight forward approach of implementing an matching engine would be the use and extension of an existing RDF query language engine, expressing clustering policies by using a RDF query language like RDQL, SeRQL, RQL or an RDF Rule language like TRIPLE.<sup>7</sup> Unfortunately none of the query engines support operators like *SIMILAR – TO* or *PART – OF – ONTOLOGY* so far. However existing concepts already shown could be used to extend such engines.

## 7 Conclusion

This paper makes several novel contributions: We introduced the concept of semantic overlay clusters in super-peer based networks. SOC's are designed for very large,

---

<sup>7</sup> We are currently evaluating how clustering rules can be mapped to the RDQL language. In a simple approach all parts of the "IF.. AND.. AND" clauses could be mapped to the AND clauses of the RDQL language. The SELECT part and the WHERE part as well as the USING part could be static, since they do not change.

highly distributed networks improving search and semantic interoperability. Especially the super-peer topology, consisting of a super-peer backbone with powerful computers and smaller clients which are linked to these super-peers, is very suitable for this approach. Further on we showed four extensions to an existing super-peer network, allowing a dynamic clustering of information provider peers to super-peer based clusters: RDF-based models for information provider peers formulated by using knowledge from existing approaches of the data base community, clustering policies expressing the demand on information providers based on existing RDF Query languages, distribution concepts models for based on the HyperCuP algorithm and finally matching approaches. Implementing the shown concepts within the Edutella network by using existing components is left open to further work.

## 8 Acknowledgements

We thank Susanne Busse and Ralf-Detlef Kutsche from CIS, TU-Berlin for reviewing concepts presented and providing the research environment.

## References

1. K. Aberer. P-grid: A self-organizing access structure for p2p information systems. In *Proceedings of the Sixth International Conference on Cooperative Information Systems (CoopIS)*, Trento, Italy, 2001.
2. 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.
3. 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.
4. 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.
5. C. Buckley, A. Singhal, M. Mitra, and G. Salton. New retrieval approaches using smart: Trec.
6. D. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2:139–172, 1987.
7. H. Galhardas, D. Florescu, D. Shasha, and E. Simon. An extensible framework for data cleaning. In *ICDE*,, page 312, San Diego, CA, 2000.
8. Gallager, Humblet, and Spira. A distributed algorithm for minimum weight spanning trees. In *ACM Transactions on Programming Languages and Systems*, volume 5-1, pages 66–77, 1983.
9. H. Garcia-Molina and B. Yang. Efficient search in peer-to-peer networks. In *Proceedings of ICDCS*, 2002.
10. L. Gong. Project JXTA: A technology overview. Technical report, SUN Microsystems, Apr. 2001. <http://www.jxta.org/project/www/docs/TechOverview.pdf>.
11. A. Y. Halevy, Z. G. Ives, P. Mork, and I. Tatarinov. Piazza: Data management infrastructure for semantic web applications. In *Proceedings of the Twelfth International World Wide Web Conference (WWW2003)*, Budapest, Hungary, May 2003.

12. M. A. Hernández and S. J. Stolfo. Real-world data is dirty: Data cleansing and the merge/purge problem. *Data Mining and Knowledge Discovery*, 2(1):9–37, 1998.
13. Jeen Broekstra et.al. A metadata model for semantics-based peer-to-peer systems. In *In Proceedings of the International Workshop in Conjunction with the WWW03 Budapest*, 2003.
14. V. Kashyap and A. Sheth. *Information Brokering Across Heterogeneous Digital Data A Metadata-based Approach*. Kluwer Academic Publishers, Boston/Dordrecht/London, 2000.
15. A. Löser, W. Nejdl, M. Wolpers, and W. Siberski. Information Integration in Schema-Based Peer-To-Peer Networks. In *Proceedings of the 15th International Conference of Advanced Information Systems Engineering (CAiSE 03)*, Klagenfurt, June 2003.
16. E. Mena, V. Kashyap, A. P. Sheth, and A. Illarramendi. OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies. In *Conference on Cooperative Information Systems*, pages 14–25, 1996.
17. S. Mohan and M. J. Willshire. DataBryte: A data warehouse cleansing framework. In *Proceedings of the International Conference on Information Quality (IQ)*, pages 77–88, Cambridge, MA, 1999.
18. F. Naumann. *Quality-driven Query Answering for Integrated Information Systems*, volume 2261 of *Lecture Notes on Computer Science (LNCS)*. Springer Verlag, Heidelberg, 2002.
19. 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 *Proceedings of the Eleventh International World Wide Web Conference (WWW2002)*, Hawaii, USA, May 2002.
20. 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.
21. C.-H. Ng, K.-C. Sia, and I. King. Peer clustering and firework query model in the peer-to-peer network. Technical report, Chinese University of Hongkong, Department of Computer Science and Engineering, 2003.
22. M. Schlosser, M. Sintek, S. Decker, and W. Nejdl. A scalable and ontology-based P2P infrastructure for semantic web services. In *Proceedings of the Second International Conference on Peer-to-Peer Computing*, Linköping, Sweden, September 2002.
23. Semantic overlay networks, November 2002. Submitted for publication.
24. H. Tangmunarunkit, S. Decker, and C. Kesselman. Ontology-based resource matching - the grid meets the semantic web. In *Proceedings of the First Workshop of Semantics in Peer-to-Peer and Grid Computing in Conjunction with the 12. th WWW Conference*, Budapest, 2003.
25. K. Thompson and P. Langley. Concept formation in structured domains. In *D. Fisher, M. Paz-zani, and P. Langley, editors, Concept formation: knowledge and experience in unsupervised learning*. Morgan Kaufmann., 1991.
26. G. Wiederhold. Mediators in the architecture of future information systems. *IEEE Computer*, 25(3):38 – 49, 1992.
27. B. Yang and H. Garcia-Molina. Designing a super-peer network. In *Proceedings of the ICDE*, March 2003.
28. S. Q. Zhuang, B. Y. Zhao, A. D. Joseph, R. H. Katz, and J. D. Kubiatowicz. Bayeux: An architecture for scalable and fault-tolerant wide-area dissemination. In *Proceedings of ACM/NOSSDAV*, Port Jefferson, New York, USA, June 2001.