Semantic Gossiping: Fostering Semantic Interoperability in Peer Data Management Systems

Karl Aberer, Philippe Cudré-Mauroux, and Manfred Hauswirth

School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland
{karl.aberer, philippe.cudre-mauroux, manfred.hauswirth}@epfl.ch

1 Introduction

Ubiquitous availability of digital equipment has transformed end-users into industrious producers of digital content. As a matter of fact, electronic devices do not only produce content, but also metadata — for example date and time when a picture was taken, technical information on the quality of a photo, or even GPS coordinates. Additionally, users often annotate their information with further metadata to increase usability. From an information systems point of view, this means that the information producers, the "domain experts", augment automatically produced metadata with high-quality domain-specific metadata at the data source.

As digital content is shared over the Internet, enormous amounts of data suddenly become available, additionally to the local data already available anyway. This proliferation of digital content calls for the development of large-scale infrastructures to share, integrate, and process massive amounts of data in a meaningful way. For example, it would be highly interesting to connect photos which are annotated with GPS coordinates to descriptions of the locations. This is just a simple, yet rather useful example, e.g., when it comes to travel planning. Many other interesting applications can be imagined. All of them, however, require the availability of metadata and their integration, as no universal metadata schema exists and is unlikely to exist ever.

Peer-to-Peer (P2P) systems already enable the efficient sharing of information on a global scale. Integration and use of metadata, however, falls short in these systems. Current P2P systems either impose a simple semantic structure a-priori (e.g., Napster or Kazaa) and leave the burden of semantic annotation and integration to the user, or do not address the issue of semantics at all (e.g., Gnutella, standard DHT-based infrastructures) but simply support a semantically unstructured data representation and leave the burden of "making sense" to the skills of the user, e.g., by providing pseudo-structured file names such as *Enterprise-2x03-Mine-Field* that encapsulate very simple semantics.

This situation exemplifies again a key problem in current Internet information systems: the lack of semantic interoperability. Semantic interoperability is a crucial element for making distributed information systems usable. It is a prerequisite for structured, distributed search, data exchange, and data integration, and provides the foundations for higher level services and processing. Classical attempts to make information resources semantically interoperable, in particular in the domain of database integration, do not scale well to global information systems such as P2P systems. Despite a large number of approaches and concepts, such as federated databases, the mediator concept [8], or ontology-based information integration approaches [5, 6], practically engineered solutions are still frequently hard-coded and require substantial support from human experts. Typical examples of such systems are domain-specific portals such as CiteSeer (*citeseer.ist.psu.edu*, publication data), SRS (*srs.ebi.ac.uk*, biology) or *streetprices.com* (e-commerce). They integrate data sources on the Internet and store them in a central warehouse. The data is converted to a common schema which usually is of simple to medium complexity. This approach adopts a simple form of wrapper-mediator architecture and typically requires substantial development efforts for the automatic or semi-automatic generation of mappings from the data sources into the central schema.

Following the P2P paradigm, new architectures addressing the issue of semantic interoperability without relying on central coordination/knowledge have recently appeared. These architectures, today commonly referred to as *peer data management systems* (PDMS), rely on pair-wise mappings (e.g., views) to foster semantic interoperability among heterogeneous information parties. In the following, we present an approach for obtaining semantic interoperability in PDMS as a result of an iterative process. We assume that users who are interested in information from other users provide mappings between their own metadata (schema) and other metadata (schemas). Also we assume that users make these pair-wise, local mappings accessible. We then build on the principle of gossiping, which has been successfully applied for creating useful global behaviors in P2P systems.

In any P2P system, search requests are routed in a network of interconnected information systems. We extend the operation of these systems as follows: When different schemas are involved, local mappings are used to further distribute a search request to other semantic domains. The quality of search results in a gossiping-based approach clearly depends on the quality of the local translations in the translation graph. To take this into account, *our fundamental assumption is that the translation links / schema mappings may be incorrect*. Thus, our mechanisms try to determine which translations can be trusted and which cannot and take this into consideration to guide the search process.

In the following we will present our model in detail and identify different methods that can be applied to estimate the quality of local mappings. We elaborate the details of each of these methods for a simple example. The information obtained from these analyses are then used by the peers to direct searches in the network of semantically heterogeneous information sources, and to iteratively correct erroneous mappings in a self-organizing way. Also, we develop heuristics to analyze semantic interoperability in large and heterogeneous communities. Finally, we present our *GridVine* system, which implements our approach and provides a semantic overlay to demonstrate how our approach can be deployed in a practical setting.

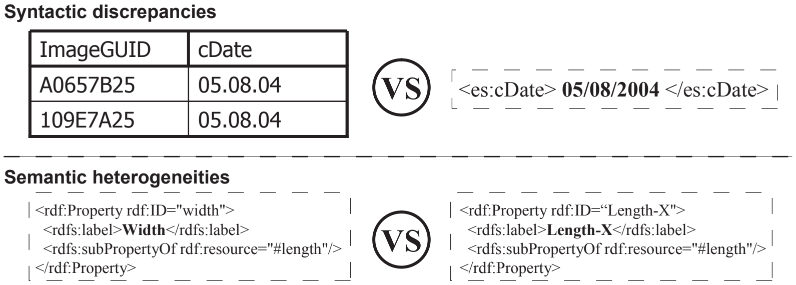
We believe that this radically new approach to semantic interoperability shifts the attention from problems that are inherently difficult to solve in an automated manner at the global level (*"How do humans interpret information models in terms of real world concepts?"*), to a problem that leaves vast opportunities for automated processing and for increasing the value of existing information sources, namely the processing of existing local semantic relationships in order to raise the level of their use from local to global semantic interoperability.

2 Motivation: Sharing Images Meaningfully

Scanners, digital cameras, webcams or new generations of cell phones: Over the last few years, we have witnessed a constant evolution and miniaturization of digital imaging equipment. Digital technologies have superseded traditional imaging technologies in most aspects of our everyday life. As a result, it is today not uncommon for end-users to store hundreds of megabytes or even gigabytes of images on their personal devices. The ways of sharing these images remain however rater primitive. Thus, most pictures remain local while only a small fraction gets shared, either through simple web galleries or point-to-point communication (e.g., SMTP or MMS). Distributed search capabilities are equally flawed, as they often revolve around keyword searches on the filename of the image (often a serial number nowadays) or take advantage of low-level features (color moments, textures) barely connected to higher-level semantics.

Digital imaging devices offer a real opportunity for creating large-scale sharing infrastructures by leveraging on local metadata production. An increasing number of approaches use metadata to organize images locally: The Extensible Metadata Platform (XMP), Adobe Photoshop Album or Microsoft WinFS are just a few examples of this new trend. The question is: Why are these local metadata not exploited in the large? Because of two fundamental hurdles (see Figure 1):

Syntactic discrepancies: Data models can differ among tools. Even for a single data model, radically different encoding schemes often coexist. Although



ImageGUID	cDate
A0657B25	05.08.04
109E7A25	05.08.04

```
<es:cDate> 05/08/2004 </es:cDate>
```

```
<rdf:Property rdf:ID="width">
  <rdfs:label>Width</rdfs:label>
  <rdfs:subPropertyOf rdf:resource="#length"/>
</rdf:Property>
```

```
<rdf:Property rdf:ID="Length-X">
  <rdfs:label>Length-X</rdfs:label>
  <rdfs:subPropertyOf rdf:resource="#length"/>
</rdf:Property>
```

Fig. 1. Two hurdles preventing the exploitation of local metadata in the large

well-documented, reconciling different formats is still a tedious and error-prone task in practice. For example, in the upper part of Figure 1, a relational representation is used on the left side while the right side uses a semi-structured model to encode the same information. Orthogonal to this issue, there is also the problem of translating items which cannot be represented in another format. This may lead to significant loss of information, especially while translating a query from one semantic domain into another (see Sec. 4).

Semantic heterogeneities: It is worth noticing that all the aforementioned metadata platforms are extensible, i.e., they all expect end-users to define their own vocabulary to describe data. In such a context, common agreements on the data model and encoding scheme cannot alone guarantee semantic interoperability. Some ontological commitment on the various concepts introduced by the peers is required to ensure meaningful communication. For example, in the lower part of Figure 1, both sides define the same concept but use different concept names. Of course, this is just a very simple example, yet it would already impair integration and query forwarding in current P2P systems.

The second problem is particularly difficult to tackle in decentralized environments like the one we are considering. Indeed, standard semantic reconciliation techniques (e.g., LAV or GAV) rely on a predefined, global schema, which would be impossible to enforce in a world-wide P2P context. In the following, we explore a different avenue by considering local schemas and schema mappings only, incrementally integrating them and guiding the integration with quality measures.

3 Of Semantic Neighborhoods and Schema Translations

Without constraining general applicability, we assume that there exists a P2P communication facility among the participants that enables sending and receiving of information, i.e., queries, data, and schema information. In the P2P system, groups of peers may have agreed on common semantics, i.e., a common schema. We denote these groups as *semantic neighborhoods.* The size of a neighborhood may range from a single individual peer up to any number. If two peers located in two disjoint neighborhoods meet, they can exchange their schemas and provide mappings between them (how peers meet and how they exchange this information depends on the underlying system but does not concern our approach). Figure 2 shows a simple mapping graph assuming each of the peers (denoted by circles) uses its own schema.

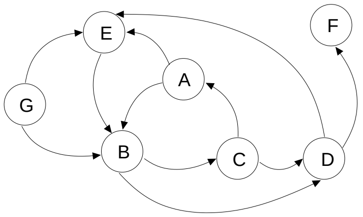


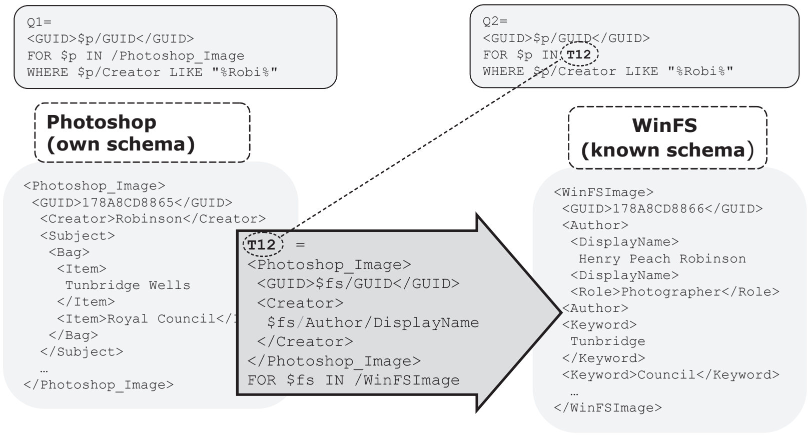
Fig. 2. Mapping graph among semantic neighborhoods

The direction of the mapping and the peer providing a mapping are not necessarily correlated. For example, peers A and B might both provide a mapping from $schema(A)$ to $schema(B)$, and they may exchange these mappings upon discretion. Figure 3 shows an example of how mappings among heterogeneous schemas can be applied to support query forwarding. In this example we use XML and XQuery as one possible way to encode the mappings. In fact, our approach works irrespective of the data model (see Sec. 7 for an application based on triples), mapping or query language used.

By providing a mapping translation ($T12$) among the Photoshop and the WinFS schema (see figure), queries against the Photoshop database ($Q1$) can also be posed against a WinFS database ($Q2$). Both the queries and the translation are expressed in XQuery. Assuming that the two databases reside at two different peers, this setup enables data integration and query forwarding.

During the life-time of the system, each peer has the possibility to learn about existing mappings and may add new ones. This means that a directed graph of mappings as shown in Figure 2 will be built between the neighborhoods along with the normal operation of the system, e.g., query processing and query forwarding in a P2P system.

The mapping graph has two interesting properties: (1) based on the already existing mappings and the ability to learn about existing mappings, new mappings can be added automatically by means of transitivity, for example, $D \to E \to B \Rightarrow D \to B$ and (2) the graph has cycles. (1) means that we can propagate queries towards peers for which no direct translation link exists. This is what we call *Semantic Gossiping.* (2) gives us the possibility to assess



```
Q1=
<GUID>$p/GUID</GUID>
FOR $p IN /Photoshop_Image
WHERE $p/Creator LIKE "%Robi%"
```

```
Q2=
<GUID>$p/GUID</GUID>
FOR $p IN T12
WHERE $p/Creator LIKE "%Robi%"
```

```
<Photoshop_Image>
 <GUID>178A8CD8865</GUID>
  <Creator>Robinson</Creator>
  <Subject>
   <Bag>
    <Item>
     Tunbridge Wells
    </Item>
    <Item>Royal Council</I
   </Bag>
  </Subject>
  …
</Photoshop_Image>
```

```
T12  =
<Photoshop_Image>
 <GUID>$fs/GUID</GUID>
 <Creator>
  $fs/Author/DisplayName
 </Creator>
</Photoshop_Image>
FOR $fs IN /WinFSImage
```

```
<WinFSImage>
 <GUID>178A8CD8866</GUID>
 <Author>
  <DisplayName>
   Henry Peach Robinson
  <DisplayName>
  <Role>Photographer</Role>
 <Author>
 <Keyword>
  Tunbridge
 </Keyword>
 <Keyword>Council</Keyword>
  …
</WinFSImage>
```

Fig. 3. A translation link using a mapping in XQuery to transform queries

the degree of *semantic agreement* along a cycle, i.e., to measure the quality of the translations and the degree of semantic agreement in a community.

In such a system, we expect peers to perform several task: (1) upon receiving a query, a peer has to decide where to forward the query to, based on a set of criteria that will be introduced below; (2) upon receiving results or feedback (cycle), it has to analyze the quality of the results at the schema and at the data level and adjust its criteria accordingly; and (3) it has to update its view of the overall semantic agreement.

The criteria to assess the quality of translations — which in turn is a measure of the semantic agreement — can be categorized as *syntactic* and *semantic*. Syntactic criteria relate only to the processed query and to the required translation. For example, by *syntactic similarity* we denote the extent to which a query is preserved after translation (lost attributes, etc.). Semantic criteria relate to the degree of agreement that can be achieved among different peers upon specific translations. Such degrees of agreement may be computed using feedback mechanisms, for example from cycles in the translation graph or from results returned. This means that a peer will locally obtain both returned queries and data through multiple cycles. In case a disagreement is detected (e.g., a wrong attribute mapping at the schema level or the violation of a constraint at the data level), the peer has to suspect that at least some of the mappings involved in the cycle were incorrect, including the mapping it has used itself to propagate the query. Even if an agreement is detected, it is not clear whether this is not accidentally the result of compensating mapping errors along the cycle. Thus, analyses are required that assess which are the most probable sources of errors along cycles, to what extent the own mapping can be trusted and therefore how to use these mappings in future routing decisions. At a global level, we can view the problem as follows: The translations among domains of semantic homogeneity (i.e., sets of related schemas) form a directed graph. Within that directed graph we find cycles. Each cycle allows to return a query to its originator which in turn can make the analysis described above.

Assuming all the peers implement this approach, we expect the network to converge to a state where a query is only forwarded to the peers that most-likely understand it ("can understand it") and where the correct mappings are increasingly reinforced by adapting the per-hop forwarding behaviors of the peers. Implicitly, this is a state where a *global agreement* on the semantics of the different schemas has been reached.

4 Semantic Query Routing

To assess the quality of schema translations we use a set of measures which guide the query forwarding process. The analysis includes (1) a syntactic assessment that covers issues such as loss of attributes, combination of attributes, etc., and (2) semantic analyses that address correctness of translations at the schema and the data level. If the combination of these measures is above a certain threshold, then a query will be forwarded over a certain link, otherwise it will be dropped. The analysis is done continuously, such that query routing paths will change dynamically over time, mirroring the situation in the network.

The query forwarding algorithm itself looks as follows:

1. when a query is first received, return potential results
2. in case the local neighborhood has not received the query, forward it to the local neighborhood
3. detect any semantic cycles and do the necessary bookkeeping
4. when a query is first received, perform the following operations for each of the outgoing links for which a translation is known:
   a) apply the translation to the query
   b) update the similarity measures for the transformed query
   c) test the similarity measures against predefined thresholds
   d) forward the query if all similarity measure tests succeed, i.e., if the transformed query can still be considered as close enough to the original query

In the following we give an informal overview of the measures we use and how to compute them. Complete mathematical definitions and derivations are given in [1].

4.1 Syntactic Measures

When forwarding a query through a translation link, parts of the query may be lost since the schema which the query is mapped onto may not have a representation for the information contained in the original schema. *Syntactic similarity* provides a measure which is related to this type of information loss during translation. This measure is context-independent since its evaluation relies exclusively on the inspection of the syntactic features of the translated queries. A high syntactic similarity does not ensure that forwarding a query is useful, but conversely a low syntactic similarity implies that it might not be useful to further forward a query.

Let us suppose we have a relational query q, originally applied to database DB_1 with schema S_1. Assume a transformation T of query q is given, such that q can be evaluated against database DB_2 with schema S_2, i.e., $T(q)(DB_2)$, which again can be given in the form of a query. T is the schema mapping we have discussed in the previous section. The problem we need to assess in analyzing the quality of T in respect to syntactic similarity is that it might occur that attributes used in q are no longer available after applying T to q, i.e., they are lost. These losses may be of varying degrees as the importance of attributes is query dependent. We have two issues to consider after applying a composite transformation $T = T_1 \circ \ldots \circ T_n$.

Not all attributes in selection predicates are preserved. In terms of SQL, these are the attributes used in the predicates of the WHERE clause of the statement. If some of these attributes cannot be mapped, some of the predicates will not be correctly evaluated (in our case, the predicates will simply be dropped). Depending on the selectivity of the predicate, this might be harmful to different degrees. We capture this by assigning selectivity values $\in [0, 1]$ to all selection predicates. High values indicate highly selective attributes, i.e., attributes whose predicates select a small proportion of the database. Thus dropping highly selective and thus more critical attributes will lead to lower the value of this measure. In conjunction with additional user-defined importance weights we can derive a syntactic similarity measure in respect to selection. The overall similarity measure combines all attribute measures and will decrease proportionally to the relative weight and selectivity of every attribute lost in the selection.

Not all projection attributes are preserved. In terms of SQL, these are the attributes used in the SELECT clause of the statement. If not all projection attributes are preserved, the results returned may be incomplete or even erroneous. Following the method used above for selection, we capture this by calculating a value $\in [0, 1]$. Again, this similarity measure combines the individual attribute measures and decreases with the number of translations applied to the query, until it reaches 0 when all the projection attributes are lost.

4.2 Semantic Measures

The context-independent measure of syntactic similarity is based on the assumption that the query transformations are semantically correct, which in general might not be the case. A better way to view semantics is to consider it as an agreement among peers. If two peers agree on the meaning of their schemas, then they will generate compatible translations. From that basic observation, we will clearly need context-dependent measures of *semantic similarity*. These measures will allow us to assess the quality of attributes that are preserved in the translations. We use two mechanisms for deriving the quality of a translation. One mechanism, *cycle analysis*, is based on analyzing the correctness of translations at the schema level, the other one, *result analysis*, is based on analyzing the quality of query results at the data level.

Cycle Analysis

For the first mechanism, we exploit the fact that in forwarding queries, circles may occur. A translation T applied to a query actually means that a peer p_1 applies a translation T to a query and forwards it to a peer p_2. For this we use the abbreviated notation $T_{p_1 \to p_2}$. A circle then simply means that there exists a sequence $T_{p_1 \to p_2}, T_{p_2 \to p_3}, \ldots T_{p_n \to p_1}$ for some $n > 1$. The returning query q_n is of the form $q_n = (T_{p_1 \to p_2} \circ T_{p_2 \to p_3} \circ \ldots \circ T_{p_n \to p_1})(q_1)$. p_1 may now analyze the differences between the original query q and the returning query q_n. It could attempt to check whether the composed transformation is identity, but the approach we propose here appears to be more practical. We inspect all attributes present in the original query distinguishing three cases:

1. The attribute is maintained throughout the cycle. This indicates that all the peers along the cycle agree on the meaning of the attribute, which increases the confidence in the correctness of the translations used (*positive feedback*).
2. The attribute is lost. This means that someone along the cycle had no representation for the attribute which is thus not part of the common semantics; This case is handled by the syntactic analyses. This leaves the semantic confidence in the translations unchanged (*neutral feedback*).
3. The attribute is mapped onto another attribute in the returning query. This indicates some semantic confusion along the cycle. Subcases can occur depending on the cases. This lowers the confidence in the translations (*negative feedback*).

Clearly p_1 may receive multiple cycle messages from different cycles and we have derived heuristics which allow p_1 to assess the correctness of the translation $T_{p_1 \to p_2}$ it has used based on the different cycle messages it receives. These heuristics also have to take into account *compensating errors*, i.e., series of independent translation errors resulting in a correct overall translation that may occur along the cycle of foreign links without being noticed by p_1, which only has the final result q_n at its disposal. Again, we come up with an overall measure whose value starts from 1 (in the semantic domain which the query originates from) and decreases as the query traverses more and more semantically heterogeneous domains. A detailed mathematical derivation of this similarity measure, along with longer explanations are given in [1].

Result Analysis

The second mechanism for analyzing the semantic quality of the translations is based on the analysis of the results returned. Although the results returned to a peer may be correct at the schema level, the data returned may not necessarily make sense. By using low-level analyses on the returned content, a peer can determine wether or not it received what it was expected to receive. Queries in our metadata model are an intensional way of expressing semantic concepts, whereas extensionally the concepts are related to sets of documents. By relating the intensional notion of a concept from the query it sent to the extensional notion of the concept as conveyed by the results it received, a peer can try to assess the semantic quality of the translation links which have been used to forward the query.

These analyses also result in positive and negative feedback which can be combined into an overall measure for the quality of the translation links. Again, we refer the interested reader to [1] for the low-level conceptual and mathematical details.

4.3 An illustrating example

To illustrate query forwarding and the application of the syntactic and semantic measures, let us consider the simple semantic network depicted in Figure 4.

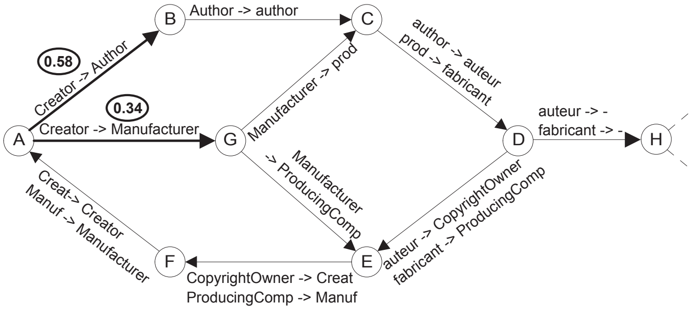


Fig. 4. A simple semantic network

The network is composed of eight peers ($A \dots H$) sharing digital pictures that have been annotated independently. Peer A wants to retrieve images based on the name of their creator. A can use two translation links to forward its query: the translation link between A and B, mapping A's *Creator* onto

B's *Author*, and the translation link between A and G, mapping *Creator* to *Manufacturer*.

A runs the analyses to assess the quality of its outgoing translation links for its *Creator* attribute. It issues a simple query which gets propagated to all the nodes except H, which has no representation for the attribute in question (thus, the syntactic similarity drops to zero and the query is not forwarded). A detects three cycles involving *Creator*: $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow A$, which yields positive feedback (as the transitive closure of the translations maps *Creator* onto itself), and $A \rightarrow G \rightarrow E \rightarrow F \rightarrow A$ and $A \rightarrow G \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow A$, which both yield negative feedback. By applying our heuristics, A gets 0.58 and 0.34 for the semantic quality measure of the first and the second mapping respectively. Clearly, there is some disagreement on the mappings involving G. Thus, A chooses to discard the second link and forwards the query through B only, where the semantics of its query are more likely to be preserved.

5 Self-Healing Semantic Networks

So far, we have applied Semantic Gossiping techniques to evaluate the quality of various mappings and selectively forward queries to groups of peers. Going one step further, we now take advantage of the semantic similarity results to attempt to detect mapping errors and incrementally correct them. In this way, Semantic Gossiping can be used to automatically refine semantic agreements in a large community of heterogeneous information parties.

We conducted series of experiments to evaluate this approach. We proceeded iteratively as follows: We construct a network of peers (representing each a semantic domain) interconnected with translation links using a small-world topology. We start with a certain percentage of erroneous translation links (i.e., erroneous mappings). For each iteration step, peers first have to randomly select one of their local attributes and send out a probe query for this attribute. Probe queries are routed irrespective of the syntactic analysis with a Time-To-Live (TTL) value indicating how many translation links they can traverse. Peers can evaluate the correctness of their outgoing translation links on the basis of the positive or negative feedback they receive or detect. Peers then attempt to modify their mappings in order to maximize the similarity results. They finally adopt the most probably correct mapping according to our analyses.

What is the result of this process in the long run? It depends of course on the initial setting but in the end, our method attempts to obtain a mapping consensus based on the feedback received from the rest of the network. Considering a high density of links and relatively few erroneous links, the method converges (i.e., repairs all erroneous mappings) rapidly, since peers can base their decisions on numerous and meaningful feedback cycles or documents.

For settings where links are scarce, peers do not have sufficient information for making sensible choices, and results may diverge.

We give below a few examples illustrating this point. Complete experimental results may be found in [1]. Figure 5 shows the sensitivity to the number of translation links l connected to each peer (cycle analysis only): the higher the number of links, the greater the number of cycles which can be detected. We start with $eRate = 10\%$ of erroneous links (Y-axis) and iteratively conduct series of cycle analyses (X-axis). For low numbers of translation links, peers simply do not get sufficient feedback information to correct mappings. For high values (e.g., 5 translation links per domain), peers receive sufficient information to correct most (or even all) of the erroneous mappings after nine iterations. Thus, the importance of the density of the translation links in a network of heterogeneous peers.

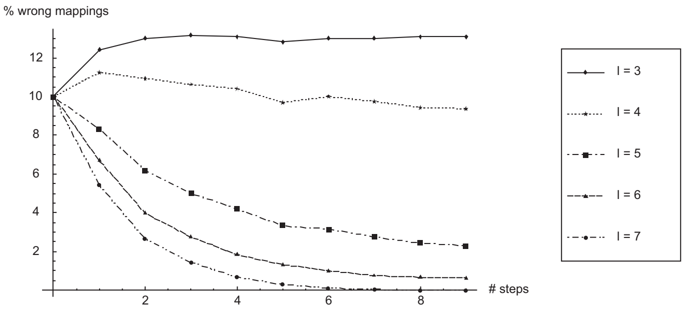


Fig. 5. Sensitivity to the number l of translation links/peer (cycle analysis only) for 25 peers, 4 attributes/peer and $TTL = 5$ hops

Figure 6 below shows some scalability results. Given that the peers apply the methods we have presented in their vicinity only and that we do not rely on any central component or computation, it is not surprising to see that the results are rather insensitive to the network size.

Combining both the cycle and the result analyses, we can push the evaluation further and apply our healing process on very faulty topologies. Figure 7 reports on an experiment where at each step, every peer first performs a result analysis step (modifying the mappings depending on the results returned) and then performs a cycle analysis step (trying to reach some local agreement on mappings based on cycle feedback). This method takes longer to converge than the two analyses applied separately; This is because the analyses keep interfering with each other until some state is reached that is consistent from both a cycle and a feedback analysis point of view.

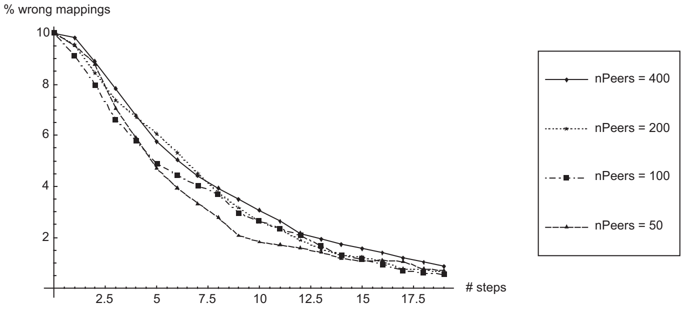


Fig. 6. Scalability (result analysis only) for 4 attributes/peer, 2 translation links/peer, $TTL = 3$ hops, 10% of misclassified documents, 2 documents/peer on average and a varying number of peers

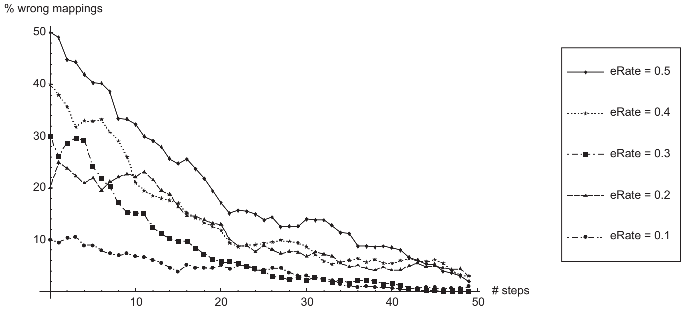


Fig. 7. Combined results for 25 peers, 4 attributes/peer and 2 translation links/peer, $TTL = 3$ hops (results), $TTL = 6$ hops (cycles), 10% of misclassified documents, 2 documents/peer on average and a varying percentage of erroneous mappings

6 Analyzing Semantic Interoperability in the Large

Considering the results from the previous section, one can observe some correlation between the topology of the network of translation links and the quality of the semantic agreement which can be achieved. This observation inspired us to develop new techniques to analyze semantic interoperability in very large networks.

Imagine a large set of peers creating and mapping schemas through translation links. The degree of interconnection of the resulting network of schemas may vary depending on the location and the number of translation links: It can be in a state where schemas are largely disconnected (subcritical state) or in a state where there are enough translation links to actually interconnect most of the schemas (supercritical state). Based on the degree distribution and some statistical aspects of the graph, and by applying a recent graph-theoretic framework, we were able to determine (see [4]) the point at which the network of translations starts to percolate, i.e., the point at which the number of translation links is such that most peers are connected to each other through transitive closures of translation links. This is highly important as a large network of peers simply cannot be semantically interoperable before this point, due to the lack of translation links.

This result also enables us to predict to what extent a query can be propagated through the P2P network. Figure 8 shows the size of the out-component—the maximal fraction of the overall network a peer is able to reach by forwarding a query through series of translation links—in a randomly generated directed graph of *10 000* heterogeneous peers (vertices) with an increasing number of translation links (edges).

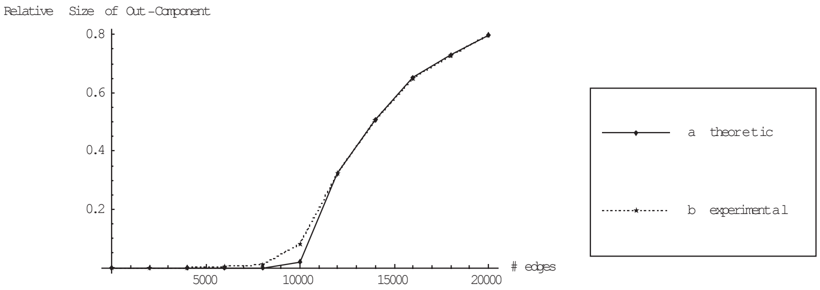


Fig. 8. Size of the out-component in a random network of *10 000* vertices

The two curves represent the relative size of the component (a) as evaluated using our graph-theoretic method and (b) as really found in the graph. As can be seen the two curves nearly perfectly match which indicates the correctness of our prediction method. As a practical application, this method can be used to measure the minimum number of translations required in a network so that it can start to self-organize and further optimize integration. It can also be used to predict an upper-bound on the number of semantic neighborhoods a query can reach.

7 GridVine: Implementing Semantic Gossiping on top of a DHT

To demonstrate the applicability of our approach, we implemented it in our GridVine [2] system. The Semantic Gossiping approach maps quite naturally onto unstructured P2P systems as they can gossip at the physical layer. In fact, most existing proposals for schema integration and query forwarding are based on some form of unstructured P2P system offering this capability. In contrast to that, we based GridVine on a structured P2P system, which is more scalable and use considerably less resources in terms of networking bandwidth. Distributed hash tables (DHTs) as Chord, CAN, Pastry or P-Grid, which are all variants of the approach suggested by Plaxton [7], are considered as the most promising candidates for next-generation P2P systems. They typically offer $\mathcal{O}(log(n))$ search complexity, with n being the number of participating peers. Although the performance and scalability of these systems are much better than that of unstructured P2P systems, they impose new problems for P2P data management approaches as broadcasting is replaced by more efficient search mechanisms.

We based GridVine on the P-Grid (http://www.p-grid.org) P2P system. In GridVine, we address the problem of building scalable semantic overlay networks by following the principle of data independence and separate the logical from the physical layer (see Figure 9).

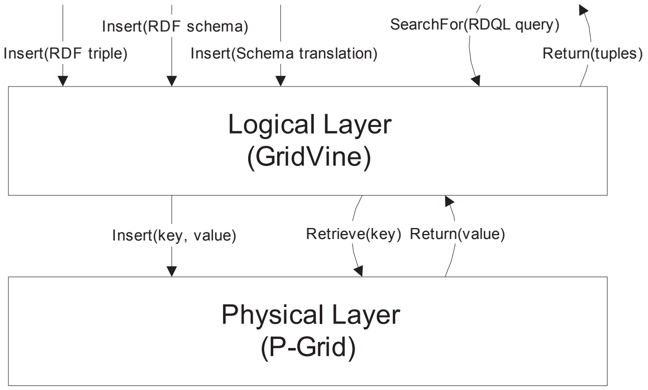
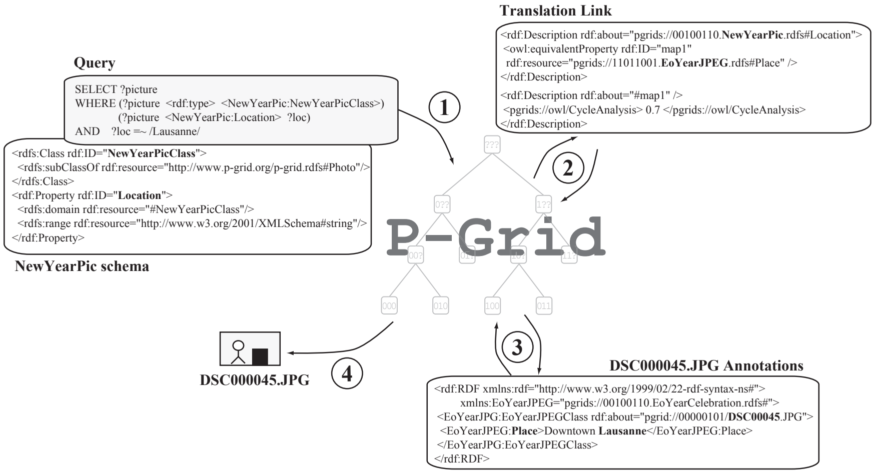


Fig. 9. GridVine: separating the logical from the physical layer

At the logical layer, we support various operations necessary for the use and maintenance of a semantic overlay network within the standard syntactic framework of RDF/OWL. We let end-users derive new schemas to annotate the content they want to share and define simple mappings to relate their schemas to other schemas. Annotations are encoded in RDF, schemas use RDFS, while translation links are expressed in OWL. All semantic data get indexed into the underlying DHT infrastructure, which supports efficient location of resources based on their identifiers (i.e., keys).

Figure 10 below gives a simplified example of query resolution in GridVine. One of the peers poses a query against a local schema called *NewYearPic* (1), as it wants to retrieve New Year's Day images taken in Lausanne. The query can be resolved in $\mathcal{O}(log(n))$ messages by routing a couple of messages through the P-Grid infrastructure, where all triples are indexed based on their *subject*, *property* and *object* values. To retrieve more results, the peer starts a Semantic Gossiping process: it searches for similar schemas, and finds a related schema, *EoYearJPEG*, with a high semantic similarity value for the property in question (2). It decides to use this schema as well and issues a transformed query, which retrieves in turn the location of a picture, *DSC000045*, whose annotations match the transformed query (3). The peer can then route a final messages through the P-Grid to retrieve the desired picture (4).



```
<rdf:Description rdf:about="pgrids://00100110.NewYearPic.rdfs#Location">
 <owl:equivalentProperty rdf:ID="map1"
  rdf:resource="pgrids://11011001.EoYearJPEG.rdfs#Place" />
</rdf:Description>
<rdf:Description rdf:about="#map1" />
 <pgrids://owl/CycleAnalysis> 0.7 </pgrids://owl/CycleAnalysis>
</rdf:Description>
```

```
SELECT ?picture
WHERE (?picture  <rdf:type>  <NewYearPic:NewYearPicClass>)
          (?picture  <NewYearPic:Location>  ?loc)
AND    ?loc =~ /Lausanne/
```

```
<rdfs:Class rdf:ID="NewYearPicClass">
  <rdfs:subClassOf rdf:resource="http://www.p-grid.org/p-grid.rdfs#Photo"/>
</rdfs:Class>
<rdf:Property rdf:ID="Location">
  <rdfs:domain rdf:resource="#NewYearPicClass"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</rdf:Property>
```

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
       xmlns:EoYearJPEG="pgrids://00100110.EoYearCelebration.rdfs#">
 <EoYearJPG:EoYearJPEGClass rdf:about="pgrid://00000101/DSC00045.JPG">
  <EoYearJPEG:Place>Downtown Lausanne</EoYearJPEG:Place>
 </EoYearJPG:EoYearJPEGClass>
</rdf:RDF>
```

Fig. 10. Resolving a query in GridVine

This kind of search process is totally automated in GridVine, as both multi-triples RDQL queries and Semantic Gossiping are supported. The separation of the physical from the logical layer allows us to process these logical operations with different physical execution strategies. In particular, we identified and implemented two alternatives for the traversal of the semantic overlay network: iterative forwarding, where the requesting peer itself repeatedly resolves and transforms the query, and recursive forwarding, where query forwarding and translation are delegated to intermediate peers.

As we rely on end-users to create schemas in GridVine, fostering interoperability among sets of independently created but semantically related schemas is crucial. To do so, we rely on schema inheritance and semantic cycle analysis. Schema inheritance enforces reuse of conceptualizations and monotonic inheritance of properties through schema hierarchies. By letting users derive subschemas from other schemas, we bootstrap semantic interoperability on small sets of properties shared by all the descendants of a (potentially very popular) base schema. Semantic cycle analysis operates as previously explained: It analyzes compositions of user-defined mappings to selectively forward queries through relevant translation links only.

To test the large-scale applicability of our approach under real-world conditions, we deployed our software on PlanetLab. PlanetLab [3] is a global testbed for large-scale experiments with distributed systems. The goal of our experiments on PlanetLab was to validate the analytical and simulation results for our semantic overlay under realistic conditions using a concrete and fully functional implementation. The results of the experiments we conducted so far match the theoretical predictions quite accurately and show that the system scales gracefully both with the number of peers and schemas.

8 Conclusions

Semantic interoperability is a key issue on the way to the Semantic Web which can push the usability of the web considerably beyond its current state. Its success, however, depends heavily on the degree of global agreement that can be achieved, i.e., on global semantics. We have presented a possible approach facilitating the fulfillment of this requirement by deriving global semantics from purely local interactions. We see our approach as a complementary effort to the on-going standardization processes in the area of semantics which may help to improve their acceptance and application by augmenting their top-down approach with a dual bottom-up strategy. We have developed our approach in a formal model that is built around a set of instruments which enable us to assess the quality of the inferred semantics. Also, we have conducted series of simulations and experiments with a concrete infrastructure legitimating our claims and provided a number of techniques aiming at a better understanding of network-related properties fostering semantic interoperability.

References

1. Aberer K, Cudré-Mauroux P, Hauswirth M (2003) — Start making sense: The Chatty Web approach for global semantic agreements. In: Journal of Web Semantics, 1(1).
2. Aberer K, Cudré-Mauroux P, Hauswirth M, van Pelt T (2004) — GridVine: Building Internet-Scale Semantic Overlay Networks. In International Semantic Web Conference (ISWC04):107–121.
3. Chun B, Culler D, Roscoe T, Bavier A, Peterson L, Wawrzoniak M, Bowman M (2003) — PlanetLab: An Overlay Testbed for Broad-Coverage Services. In: ACM SIGCOMM Computer Communication Review, 33(3).
4. Cudré-Mauroux P, Aberer K (2004) — A Necessary Condition For Semantic Interoperability In The Large. In International Conference on Ontologies, DataBases, and Applications of Semantics for Large Scale Information Systems (ODBASE04):859–872.

5. Hull R (1997) — Managing Semantic Heterogeneity in Databases: A Theoretical Perspective. In Symposium on Principles of Database Systems (PODS97):51–61.
6. Ouksel A, Ahmed I (1999) — Ontologies are not the Panacea in Data Integration: A Flexible Coordinator to Mediate Context Construction. In Distributed and Parallel Databases, 7(1):7–35.
7. Plaxton C, Rajaraman R, Richa A (1997) — Accessing Nearby Copies of Replicated Objects in a Distribute Environment In ACM Symposium on Parallel Algorithms and Architectures (SPAA97): 311-320.
8. Wiederhold G (1992) — Mediators in the Architecture of Future Information Systems. In IEEE Computer, 25(3):38–39.