Start making sense: The Chatty Web approach for global semantic agreements*

Karl Aberer†, Philippe Cudré-Mauroux, Manfred Hauswirth
School of Computer and Communication Sciences
École Polytechnique Fédérale de Lausanne (EPFL)
CH-1015 Lausanne, Switzerland

*The work presented in this paper was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

†Corresponding author. Fax: +41-21-693 8115. E-mail address: Karl.Aberer@epfl.ch

Abstract

This paper describes a novel approach for obtaining semantic interoperability in a bottom-up, semi-automatic manner without relying on pre-existing, global semantic models. We assume that large amounts of data exist that have been organized and annotated according to local schemas. Seeing semantics as a form of agreement, our approach enables the participating data sources to incrementally develop global agreements in an evolutionary and completely decentralized process that solely relies on pair-wise, local interactions.

Keywords: Semantic integration, semantic agreements, self-organization

1 Introduction

The recent success of peer-to-peer (P2P) systems and the initiatives to create the Semantic Web have emphasized again a key problem in information systems: the lack of semantic interoperability. Semantic interoperability is a crucial element for making distributed information systems usable. It is prerequisite for structured, distributed search and data exchange and provides the foundations for higher level (web) services and processing.

For example, the technologies that are currently in place for P2P file sharing systems either impose a simple semantic structure a-priori (e.g., Napster, Kazaa) and leave the burden of semantic annotation to the user, or do not address the issue of semantics at all (e.g., the current web, Gnutella, Freenet) 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.

Also, 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 [32], or ontology-based information integration approaches [12, 24], practically engineered solutions are still frequently hard-coded and require substantial support from human experts. A typical example of such systems are domain-specific portals such as CiteSeer (www.researchindex.com, 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 global schema.

In the context of the Semantic Web, a major effort is devoted to the provision of machine processable semantics expressed in meta-models such as RDF, OIL [7], OWL [5], DAML+OIL [11] and TRIPLE [28] and based on shared ontologies. Still, these approaches rely on common ontologies, to which existing information sources can be related by proper annotation. This is an extremely important development, but its success will heavily rely on the wide standardization and adoption of common ontologies or schemas.

The advent of P2P systems, however, introduces a different view on the problem of semantic interoperability by taking a social perspective which relies on self-organization heavily. We argue that we can see the emerging P2P paradigm as an opportunity to improve semantic interoperability rather than as a threat, in particular in revealing new possibilities on how semantic agreements can be achieved. This motivated us to look at the problem from a different perspective and has inspired the approach presented in this paper.

In the following, we abstract from the underlying infrastructure such as federated databases, web sites or P2P systems and regard these systems as graphs of interconnected data sources. For simplicity, but without constraining the general applicability of the presented

concepts, we denote these data sources as *peers*. Each peer offers data which are organized according to some schema expressed in a data model, e.g., relational, XML, or RDF. Among the peers, communication is supported via suitable protocols and architectures, for example, HTTP, SOAP or JXTA.

The first issue to observe is that semantic interoperability is always based on some form of agreement. Ontology-oriented approaches in the Semantic Web represent this agreement *explicitly* through a shared ontology. In our approach, no explicit representation of a globally shared agreement will be required, but agreements are *implicit* and result from the way our (social) mechanism works.

We impose a modest requirement on establishing agreements by assuming the existence of local agreements provided as partial translations between different schemas, i.e., agreements established in a P2P manner. These agreements will have to be established in a manual or semiautomatic way since in the near future we do not expect to be able to fully automate the process of establishing semantic translations even locally. However, a rich set of tools is getting available to support this [18, 23, 27]. Establishing local agreements is a less challenging task than establishing global agreements by means of global schemas or shared ontologies. Once such agreements exist, we establish on-demand relationships among schemas of different information systems that are sufficient to satisfy information processing needs such as distributed search.

We briefly highlight two of the application scenarios that convinced us (besides the obvious applicability for information exchange on the web) that enabling semantic interoperability in a bottom-up way driven by the participants is valid and applicable: introduction of meta-data support in P2P applications and support for federating existing, loosely-coupled databases.

Imposing a global schema for describing data in P2P systems is almost impossible, due to the decentralization properties of such systems. It would not work unless all users conscientiously follow the global schema. Here our approach would fit well: We let users introduce their own schemas which best meet their requirements. By exchanging translations between these schemas, the peers can incrementally come up with an implicit "consensus schema" which gradually improves the global search capabilities of the P2P system. This approach is orthogonal to the existing P2P systems and could be introduced basically into all of them.

The situation is somewhat similar for federating existing loosely-coupled databases. Such large collections of data exist, for example, for biological or genomic databases. Each database has a predefined schema and possibly some translations may already be defined between the schemas, for example data import/export facilities. However, global search, i.e., propagation of queries among the set of databases, is usually not provided and if this feature exists, it is usually done in an ad-hoc, non-systematic way, i.e., not reusable and not automated. The more complex these database schemas get, the less likely it is that the schemas partially overlap and the harder it gets to increasingly generate translations automatically.

Adopting a P2P approach is (usually) motivated by solving scalability problems. Which scalability problem are we looking at? Considering the two examples given, we observe that in both cases we face a large number of different schemas, where the interoperable schemas themselves are of modest complexity. In the case of document sharing (e.g., music files or images) the schemas are used to annotate the media content and are typically fairly simple. This is even true for media annotation in more professional settings, such as with MPEG-7 [19]. In the case of scientific data sharing the individual schemas may be fairly complex, however, the shared views typically are much simpler as the databases are very specialized on a specific problem and the "semantic intersection" among the databases is fairly small. Thus our work aims at solutions that scale well in large numbers of schemas and participants. We believe this is a critical and very realistic problem in making today's Web semantically interoperable. Our work is orthogonal to efforts in ontology engineering which are devoted to the management of one or a few large and complex ontologies, which

scale well in large numbers of concepts and rules and where social interaction occurs as part of collaborative ontology engineering [30].

In our approach, we build on the principle of gossiping that 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 into other semantic domains.

For simplicity but without constraining general applicability, we will limit the following discussions to the processing of search requests. The quality of search results in a gossiping-based approach depends clearly on the quality of the local translations in the translation graph. *Our fundamental assumption is that these translations may be incorrect*. Thus our agreement construction mechanisms try to determine which translations can be trusted and which not and take this into account to guide the search process.

A main contribution of the paper is to identify different methods that can be applied to estimate the quality of local translations from information obtained from the peer network. We elaborate the details of each of these methods for a simple data model, that is yet expressive enough to cover many practical cases (Section 3). This model is similar to other data models currently considered for semantic annotation in P2P architectures [15]. The methods that will be introduced are:

1. A syntactic analysis of search queries after transformations have been applied in order to determine the potential information-loss incurred through the transformation. Here we analyze to which degree query constituents essential for obtaining useful query results are preserved during transformation (Section 4).
2. A semantic analysis of composite translations along cycles in the translation graph, in order to determine the level of agreement that peers achieve throughout the cycle. Here we analyze whether cyclic translations preserve semantics. If concepts are not preserved in a cyclic translation we assume semantic confusion has occurred (Section 5.1).
3. A semantic analysis of search results obtained through composite translation. We assume that structured data is used to annotate media content and that peers can classify their documents both using content analysis and metadata-based classification rules. From that peers derive to which degree transformed metadata annotations match the actual content and thus how reliable the translations were (Section 5.2).

The information obtained by applying these different analyses is then used to direct searches in a network of semantically heterogeneous information sources (e.g, on top of a P2P network).

Finally we give first results that take our approach one step further. Rather than only guiding searches by the results obtained from analyzing the transformations, we also modify the translations in an automatic manner using this information (Section 7). Thus we make a step towards a self-learning network of peers automatically establishing semantic interoperability. We give experimental results that demonstrate how the different kinds of semantic analyses of mappings interact with the modification of incorrect translations and how this approach scales in different parameters.

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. The remaining problem of establishing semantic interoperability at a local level seems to be much easier to tackle once an approach such as ours is in place.

2 Overview

Before delving into the technical details, this section provides an informal overview of our approach and of the paper.

We assume that there exists a communication facility among the participants that enables sending and receiving of information, i.e., queries, data, and schema information. This assumption does not constrain the approach, but emphasizes that it is independent of the system it is applied to. The underlying system could be a P2P system, a federated database system, the web, or any other system of information sources communicating via some communication protocol. We denote the participants as peers abstracting from the concrete underlying system.

In the 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 translations between them. How peers meet and how they exchange this information depends on the underlying system but does not concern our approach. We assume that skilled experts supported by appropriate translation tools provide the translations. Later, we will also devise possibilities of how our approach might be used to automatically improve the quality of pre-existing translations by modifying them. The direction of the translation and the peer providing a translation are not necessarily correlated. For instance, peers p_1 and p_2 might both provide a translation from schema S_{p_1} to schema S_{p_2}, and they may exchange this translation upon discretion. During the life-time of the system, each peer has the possibility to learn about existing translations and add new ones. This means that a directed graph of translations as shown in Fig. 1 will be built between the peers along with the normal operation of the system (e.g., query processing and forwarding in a P2P system).

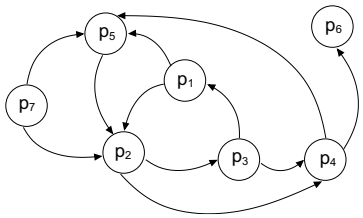

Figure 1: Translation graph among peers

This translation graph has two interesting properties: (1) based on the already existing translations and the ability to learn about existing translations, queries can be propagated to peers for which no direct translation link exists by means of transitivity, for example $p_4 \rightarrow p_5 \rightarrow p_2 \Rightarrow p_4 \rightarrow p_2$ and (2) the graph has cycles, for example $p_4 \rightarrow p_5 \rightarrow p_2 \rightarrow p_4$. We call (1) *semantic gossiping*. (2) gives us the possibility to assess 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; (2) upon receiving results or feedback along translation cycles, it has to analyze the quality of the results at the schema and at the data level and adjust its criteria accordingly; and (3) update its view of the overall semantic agreement by modifying its query forwarding criteria or by adjusting the translation themselves.

The criteria to assess the quality of translations—which in turn is a measure of the degree of semantic agreement—can be categorized as *context-independent* and *context-dependent*. Context-independent criteria, discussed in Section 4, are syntactic in nature and relate only to the transformed query and to the required translation. We introduce

the notion of *syntactic similarity* to analyze the extent to which a query is preserved after transformation.

Context-dependent criteria, which are discussed in Section 5, 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. We will introduce two such feedback mechanisms, namely cycles appearing in the translation graph and results returned by different peers. This means that a peer will locally obtain both returned queries and data through multiple feedback cycles. In case a disagreement is detected (e.g., a wrong attribute mapping at the schema level or a concept mismatch at the content level), the peer has to suspect that at least some of the translations involved in the cycle were incorrect, including the translation 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 a cycle. Thus, analyses are required that assess which are the most probable sources of errors along cycles, to what extent the own translation can be trusted and therefore of how to use these translations in future routing decisions. At a global level, we can view the problem as follows: The translations between domains of semantic homogeneity (same 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.

Each of these criteria is applied to the transformed queries and results in a *feature vector*. The decision whether or not to forward a query using a translation link then is based on evaluating these feature vectors. The details of the query forwarding process are provided in Section 6.

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 most-likely understanding it, where the correct translations are increasingly reinforced by adapting the per-hop forwarding behaviors of the peers and where incorrect translations are rectified. Implicitly, this is a state where a global agreement on the semantics of the different schemas has been reached. To demonstrate this, we present experimental results where semantic agreement is reached in a network of partially erroneous translations in Section 7.

3 The Model

3.1 The Data Model

We assume that each peer p is maintaining its database DB_p according to a schema S_p. The peers are able to identify their schema, either by explicitly storing it or by keeping a pseudo unique schema identifier, obtained for example by hashing. The schema consists of a single relational table R, i.e., the data that a peer stores consists of a set of tuples $t_1, \ldots, t_r$ of the same type. The attributes have complex data types and NULL-values are possible.

We do not consider more sophisticated data models to avoid diluting the discussion of the main ideas through technicalities related to mastering complex data models. Moreover, many practical applications, in particular in P2P systems and scientific databases, use exactly the type of simplistic data model we have introduced, at least at the meta-data level.

We use a query language for querying and transforming databases. The query language consists of basic relational algebra operators since we do not care about the practical encoding, e.g., in SQL or XQuery. The relational operators that we require are:

- Selection $\sigma_{pred(a)}(R)$, where a is a list of attribute names $A_1, \ldots, A_k$, and $pred$ is any predicate on the attributes a using standard atomic predicates on the respective datatypes, i.e., $pred = pred(A_1, \ldots, A_k)$.
- Projection $\pi_a(R)$, where a is a list of attribute names $A_1, \ldots, A_k$.

- Mapping $\mu_f(R)$, where f is a list of functions of the form $A_0 := F(A_1, \ldots, A_k)$ and $A_1, \ldots, A_k$ are attribute names occurring in R. The function F is specific to the datatypes of the attributes $A_1, \ldots, A_k$. A special case is renaming of an attribute: $A_0 := A_1$.

We assume that queries can be evaluated against any database irrespective of its schema. Predicates containing attributes not present in the evaluated schema are ignored.[1] Projection attributes which are not present in the current schema return a NULL-value. Mappings applied to non-existing attributes also return NULL-values.

3.2 The Network Model

Let us now consider a set of peers P. Each peer $p \in P$ has a basic communication mechanism that allows it to establish connection with other peers. Without loss of generality, we assume in the following that it is based on the Gnutella protocol [4]. Thus peers can send $ping$ messages and receive $pong$ messages in order to learn about the network structure. In extension to the Gnutella protocol, peers also send their schema identifier as part of the $pong$ message.

Every peer p maintains a neighborhood $N(p)$ selected from the peers that it identified through $pong$ messages. The peers in this neighborhood are distinguished into those that share the same schema, $N_e(p)$, and those that have a different schema, $N_d(p)$ as shown in Fig. 2.

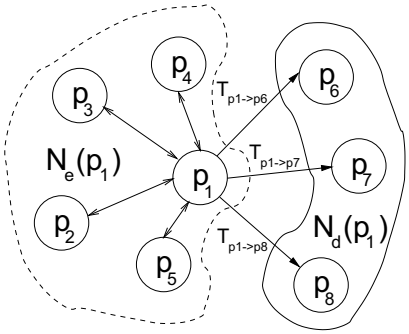

Figure 2: The network model

A peer p_1 includes another peer p_2 with a different schema into its neighborhood if it knows a transformation for queries against its own schema to queries against the foreign schema. The query transformation operator $T_{p_1 \to p_2}$ is given as a query q_T that provides a view of schema S_{p_2} according to schema S_{p_1}. In other words, q_T takes data structured according to schema S_{p_2} and transforms it into data structured according to schema S_{p_1}.

Using q_T the transformed form of a query q against a database according to schema S_{p_1} is given by $T_{p_1 \to p_2}(q)$, which is defined as

$$T_{p_1 \to p_2}(q)(DB_{p_2}) = q(q_T(DB_{p_2})).$$

We assume that translations only use a mapping operator followed by a projection on the attributes that are preserved. Thus q_T will always be of the form

$$q_T(DB_{p_2}) = \pi_a(\mu_f(DB_{p_2})).$$

Furthermore, we assume that the transformation query is normalized as follows: If an attribute A is preserved, it also occurs in the mapping operator as an identity mapping, i.e., $A := A \in f$. This simplifies our subsequent analysis.

[1] We do not use the same conventions as XPath/XQuery here, but we will make use of additional mechanisms for dropping queries.

Note that multiple transformations may be applied to a single query q. The composition of multiple transformations $T_1, \ldots, T_n$ is given by using the associative composition operator $\circ$ as follows

$$(T_1 \circ \ldots \circ T_n)(q)(DB) = q(q_{T_1} \ldots (q_{T_n}(DB))).$$

Such query transformations may be implemented easily using various mechanisms, for example XQuery as explained below.

Queries can be issued to any peer through a query message. A query message contains a query identifier id, the (potentially transformed) query q, the query message originator p, and the translation trace TT to keep track of the translations already performed. In the subsequent sections we will extend the contents of the query message in order to implement a more intelligent control of query forwarding. The basic query message format is

$$query(id, q, p, TT).$$

The translation trace TT is a list of pairs $\{(p_{from}, S_{p_{from}}), (p_{to}, S_{p_{to}})\}$ keeping track of the peers having sent the request through a translation link (p_{from}) and of the peers having received it after the translation link (p_{to}), along with their respective schema identifiers ($S_{p_{from}}$ and $S_{p_{to}}$). We will call p_{from} the sender, and p_{to} the receiver. For any translation link, we have to record both the sender and the recipient, as after a translation a query might be forwarded without transformation to peers sharing the same schema.

3.3 Case Study

To illustrate how to apply the abstract model detailed above in a concrete setting, we will now describe one of the experiments which were conducted in our group in order to realize Semantic Gossiping in an XML/XQuery environment. Note that this example will also be used in the following text to illustrate the techniques we will apply to control query propagation.

Seven people from our group were first asked to design a simple XML document containing some project meta-data. The outcome of this deliberately imprecise task definition was a collection of structured documents lacking common semantics though overlapping partially for a subset of the embraced meta-data (e.g., *name of the project* or *start date*). Viewing these documents as seven distinct semantic domains in a decentralized setting, we then produced a graph connecting the different domains together with series of translation links. The resulting topology is depicted in Fig. 3. In this figure we provide also one example of how an attribute gets transformed by the user-defined translations. All the domains have some representation for the title of the project (usually referred to as *name* or *title*, see Fig. 3 where the translations for the attribute *title* are represented on top of the links), except p_3 which only considers a mere *ID* for identifying the projects.

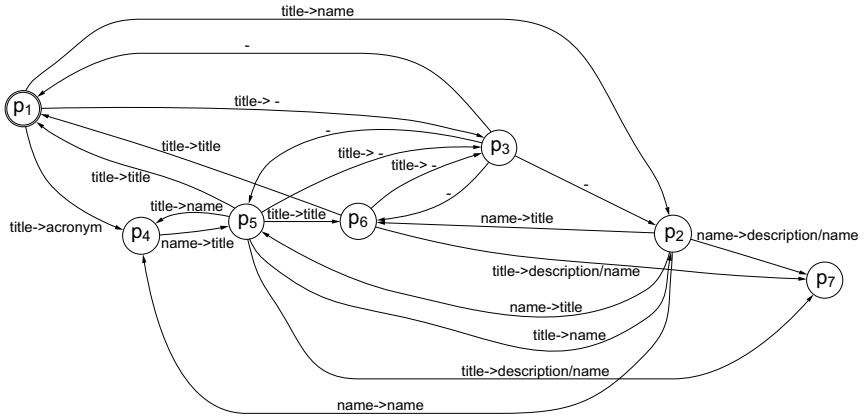

Figure 3: A semantic graph of translations

Translations were formulated as XQuery expressions in such a way that they strictly adhere to the principles stipulated above.

In the next step of the experiment, we asked the authors to write translations for every link departing from their domain (for example, p_1 was asked to provide us with the translation to p_2, p_3 and p_4). Finally, using the IPSI-XQ XQuery libraries [8] and the Xerces [26] XML parser, we built a query translator capable of handling and forwarding the queries following the gossiping algorithm. As an example for the outcome, Fig. 4 presents two different documents as well as a simple query transformation using query *T12* for translation.

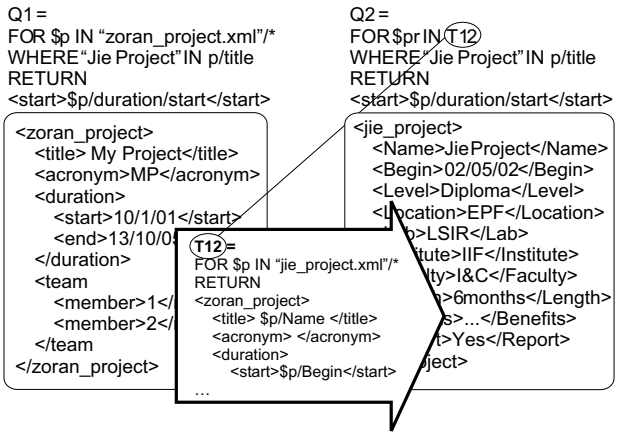

Figure 4: An example of translation mechanism

4 Syntactic Similarity

During translation, parts of queries may be lost since the schema which the query is mapped to may not have a representation for the information contained in certain attributes of 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 will 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 query q, originally applied to database DB_1 with schema S_1, which always has the generic form of a selection-projection-mapping query

$$q(DB_1) = \pi_{ap}(\sigma_{pred(as)}(\mu_{fa}(DB_1))),$$

where as is a list of attributes used in the selection predicates, ap is a list of attributes used in the projection, and fa is a list of functions applied. Without loss of generality, we assume that the query is normalized such that all attributes required in as and ap are computed by one of the functions in fa.

Assume a transformation T of query q is given, such that q can be evaluated against database DB_2 with schema S_2. The transformation is specified by a query q_T defining a view on DB_2

$$q_T(DB_2) = \pi_{ap_T}(\mu_{fa_T}(DB_2)).$$

The transformed query $T(q)$ that can be evaluated against the schema S_2 is of the form

$$T(q)(DB_2) = \pi_{ap}(\sigma_{pred(as)}(\mu_{fa}(\pi_{ap_T}(\mu_{fa_T}(DB_2))))).$$

This form will also be achieved after multiple transformations after normalization.

It might occur that attributes used in q are no longer available after applying transformation T to q. This happens when an attribute from S_2 required for the derivation of an attribute from S_1 by means of one of the functions in fa and occurring in ap or as is missing, i.e., not occurring in ap_T, or is not computed by one of the functions from fa_T.

We now determine which attributes are needed in order to properly evaluate the query q. For an attribute $A \in ap$ resp. $A \in as$ we define $source_T(A)$ as the set of attributes required in schema S_2 of database DB_2 in order to derive A by means of transformation T. If attribute A cannot be derived we will set $source_T(A) = \perp$. For a composite transformation $T_1 \circ T_2$ we have the following criterion: if $source_{T_1}(A) = \{A_1, \ldots, A_k\}$ and for all $i = 1, \ldots, k$ there exists $F_i \in fa_{T_2}$ such that $A_i = F_i(A_1^i, \ldots, A_{k_i}^i)$ then

$$source_{T_1 \circ T_2}(A) = \bigcup_{i=1,\ldots,k} \{A_1^i, \ldots, A_{k_i}^i\}.$$

If $source_{T_1}(A) = \perp$ or for some A_i no derivation of the attribute using a function $F_i \in fa_{T_2}$ is possible we have

$$source_{T_1 \circ T_2}(A) = \perp.$$

In order to ground the definition we assume that $source_\epsilon(A) = \{A\}$ and $\epsilon \circ T = T$ for the empty sequence of transformations ϵ.

In order to determine the effects of multiple transformations $T_1, \ldots, T_n$ we have to evaluate $source_{T_1 \circ \ldots \circ T_n}(A)$. This allows to determine which of the required attributes for evaluating a query containing attribute A are available after applying the transformations $T_1, \ldots, T_n$. The definition of $source$ is given such that it can be evaluated locally, i.e., for each transformation step in an iterative manner. Using this information we can now define the syntactic similarity between a transformed query and its corresponding original query.

The decision on the importance of attributes is query dependent. We have two issues to consider after applying a composite transformation $T = T_1 \circ \ldots \circ T_n$:

1. Not all attributes in as are preserved. Therefore some of the atomic predicates in $p(as)$ will not be correctly evaluated, i.e., the atomic predicates will simply be dropped in this case. Depending on the selectivity of the predicate this might be harmful to different degrees. We capture this by calculating a value FV_i^σ for every attribute $A_i \in as \cup ap$ as follows: if $A_i \in as$ and $source_T(A_i) \neq \perp$ then $FV_i^\sigma = sel_{A_i}$ else $FV_i^\sigma = 0$, where sel_{A_i} is the selectivity of an attribute A_i. The selectivity is ranging over the interval $[0, 1]$, with high values indicating 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 values of FV_i^σ

2. Not all attributes in ap are preserved. Therefore, some of the results may be incomplete or even erroneous (due to the loss of key attributes, for example). Following the method used above for the selection, we capture this by calculating a value FV_i^π for every attribute $A_i \in as \cup ap$ as follows: if $A_i \in ap$ and $source_T(A_i) \neq \perp$ then $FV_i^\pi = 1$ else $FV_i^\pi = 0$.

Given the values FV_i^σ for $A_i \in as \cup ap$ we introduce feature vectors $\overrightarrow{FV^\sigma}$ capturing the syntactic effects for the transformed query $(T_1 \circ \ldots \circ T_n)(q)$.

$$\overrightarrow{FV^\sigma}((T_1 \circ \ldots \circ T_n)(q)) = (FV_1^\sigma, \ldots, FV_k^\sigma).$$

Using this feature vector we define a syntactic similarity measure with respect to selection including a user-defined weight vector $\overrightarrow{W} = (W_1, \ldots, W_k)$ pondering the importance of the attributes as:

$$SIM_\sigma(q, (T_1 \circ \ldots \circ T_n)(q)) = \frac{\overrightarrow{W} \cdot \overrightarrow{FV^\sigma}}{|\overrightarrow{W}|\,|\overrightarrow{FV^\sigma}|}$$

where

$$\overrightarrow{W} \cdot \overrightarrow{FV^\sigma} = W_1 FV_1^\sigma + \ldots + W_k FV_k^\sigma$$

and

$$|\overrightarrow{X}| = \|\overrightarrow{X}\|_2 = \sqrt{x_1^2 + \ldots + x_k^2}.$$

This value is normalized on the interval $[0, 1]$. Originally, the similarity will be one, and it will decrease proportionally to the relative weight and selectivity of every attribute lost in the selection operator, until it reaches 0 when all attributes are lost.

For projection using the values FV_i^π the analogous feature vectors $\overrightarrow{FV^\pi}$ and similarity measures SIM_π are derived. Again, this similarity decreases with the number of translations applied to the query, until it reaches 0 when all the projection attributes are lost.

We illustrate the concepts introduced for syntactic similarity by means of a small example. Assume a peer p_1 is connected to peers p_2 and p_3 through translations as illustrated in Fig. 5.

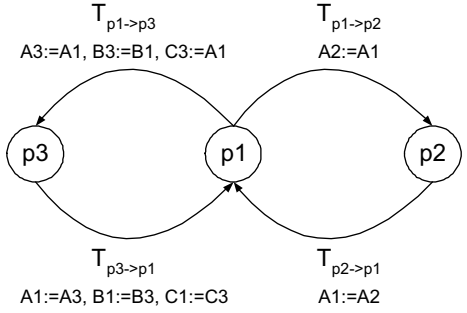

Figure 5: An example for syntactic similarity

A translation, such as $T_{p_1 \to p_3}$ can be specified as a query, e.g.,

$$q_{T_{p1 \to p3}}(DB_3) = \mu_{A_3:=A_1, B_3:=B_1, C_3:=A_1}(DB_3).$$

p_1 sends a query $q = \pi_{A_1, B_1, C_1}(DB_1)$ to the two other peers. Peer p_2 would evaluate $\overrightarrow{FV^\pi}(T_{p_1 \to p_2}(q))$ as follows: $source_{T_{p_1 \to p_2}}(A_2) = \{A_1\}$ and $source_{T_{p_1 \to p_2}}(B_2) = source_{T_{p_1 \to p_2}}(C_2) = \perp$. Therefore $\overrightarrow{FV^\pi}(T_{p_1 \to p_2}(q)(DB_2)) =$

$(1,0,0)$ and $SIM_\pi(q, T_{p_1 \to p_2}(q)) = \frac{1}{\sqrt{3}}$, assuming all user-defined weights are 1. If p_2 sends q back to p_1, p_1 would obtain $SIM_\pi(q, (T_{p_1 \to p_2} \circ T_{p_2 \to p_1})(q)) = \frac{1}{\sqrt{3}}$, since only attribute A_1 remains intact after the two translations.

On the other hand, p_3 determines $source_{T_{p_1 \to p_3}}(A_3) = \{A_1\}$, $source_{T_{p_1 \to p_3}}(B_3) = \{B_1\}$, and $source_{T_1}(C_3) = \{A_1\}$. Thus, $\overrightarrow{FV^\pi}(T_{p_1 \to p_3}(q)(DB_3)) = (1,1,1)$ and $SIM_\pi(q, T_{p_1 \to p_3}(q)) = 1$. If p_3 sends the query back to p_1, p_1 would as well obtain $SIM_\pi(q, (T_{p_1 \to p_3} \circ T_{p_3 \to p_1})(q)) = 1$. The fact that an obvious mistake occurs, i.e., that attribute $C3$ is wrongly mapped onto $A1$ in the translation, is not detected by the syntactic similarity measure, and will be dealt with by the semantic similarity measures introduced in the next section.

5 Semantic Similarity

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 now derive context-dependent measures of semantic similarity. These measures will allow us to assess the quality of attributes that are preserved in the translation.

To that end, we introduce two mechanisms for deriving the quality of a translation. One mechanism will be based on analyzing the fidelity of translations at the schema level, the other one will be based on analyzing the quality of the correspondences in the query results obtained at the data level.

5.1 Cycle Analysis

For the first mechanism, we exploit the protocol property that detects cycles as soon as a query reenters a semantic domain it has already traversed (see Section 6.1 for more details). A cycle starts with a peer p_1 transmitting a query q_1 to a peer p_2 through a translation link $T_{p_1 \to p_2}$ (see Fig. 6).

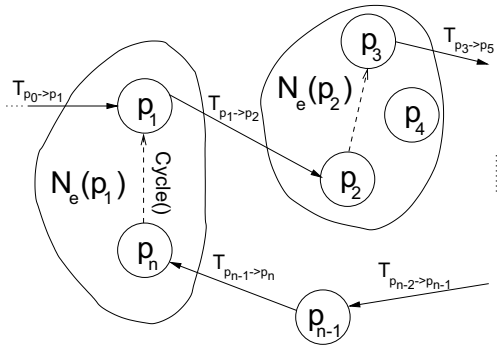

Figure 6: The feedback mechanism

In the example, after a few hops, the query is finally sent to a peer p_n which, sharing the same schema as p_1, detects a cycle and informs p_1. The returning query q_n is of the form

$$q_n = (T_{p_1 \to p_2} \circ T_{p_3 \to p_5} \circ \ldots \circ T_{p_{n-1} \to p_n})(q_1) = T(q_1).$$

p_1 may now analyze what happened to the attributes $A_1 \ldots A_k$ originally present in q_1. It could attempt to check whether the composed transformation is identity, but the approach we propose here appears more practical. We differentiate three cases:

– Case 1: $source_T(A_i) = \{A_i\}$, this means that A_i has been maintained throughout the cycle. It usually indicates that all the peers along the cycle agree on the meaning of the attribute. Such an observation increases the confidence in the correctness of the translations used.
– Case 2: $source_T(A_i) = \perp$, this means that someone along the cycle had no representation for A_i. A_i is not part of the common semantics. This leaves the confidence in the translations unchanged.
– Case 3: Otherwise, if none of the two previous cases occurs, e.g., $source_T(A_i) = \{A_j\}, j \neq i$, this indicates some semantic confusion along the cycle. Subcases can occur depending on what happens to A_j. This lowers the confidence in the translations.

We now derive heuristics for 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 received. Let us consider a translation cycle f composed of $\|f\|$ translation links. On an attribute basis, f may result in *positive* feedback (case 1 above), *neutral* feedback (case 2, not used for the rest of this analysis but taken into account by the syntactic similarity), or *negative* feedback (case 3). We denote by ϵ_{cyc} the probability of a foreign translation (i.e., $T_{p_3 \to p_5} \ldots T_{p_{n-1} \to p_n}$) along a cycle being wrong for the attribute in question. Considering these error probabilities as being independent and identically distributed random variables, the probability of not having a foreign translation error along the cycle is

$$(1 - \epsilon_{cyc})^{\|f\|-1}.$$

Moreover, *compensating errors*, i.e., series of independent translation errors resulting in a correct translation, 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. Thus, assuming $T_{p_1 \to p_2}$ correct and denoting by δ_{cyc} the probability of errors being compensated somehow, the probability of a cycle being positive is

$$(1 - \epsilon_{cyc})^{\|f\|-1} + (1 - (1 - \epsilon_{cyc})^{\|f\|-1})\delta_{cyc} = prob^+(\|f\|, \epsilon_{cyc}, \delta_{cyc}) \quad (1)$$

while, under the same assumptions, the probability of a cycle being negative is

$$(1 - (1 - \epsilon_{cyc})^{\|f\|-1})(1 - \delta_{cyc}) = 1 - prob^+(\|f\|, \epsilon_{cyc}, \delta_{cyc}). \quad (2)$$

Similarly, if we assume $T_{p_1 \to p_2}$ to be incorrect, the probability of a cycle being respectively negative and positive are

$$(1 - \epsilon_{cyc})^{\|f\|-1} + (1 - (1 - \epsilon_{cyc})^{\|f\|-1})(1 - \delta_{cyc}) = prob^-(\|f\|, \epsilon_{cyc}, \delta_{cyc}) \quad (3)$$

and

$$(1 - (1 - \epsilon_{cyc})^{\|f\|-1})\delta_{cyc} = (1 - prob^-(\|f\|, \epsilon_{cyc}, \delta_{cyc})). \quad (4)$$

Assume a peer p_1 obtains a set of positive and negative feedbacks along cycles $F = \{f_1, \ldots, f_m\}$ of lengths $\|f_1\|, \ldots, \|f_m\|$ for a given attribute A. Some of these may be positive, i.e., $source_T(A) = \{A\}$, other negative. We denote by $F^+ \subseteq F$ the set of positive and by $F^- \subseteq F$ the set of negative feedbacks and have $F = F^+ \cup F^-$.

If p_1 assumes that its own outgoing translation link at the start of the cycle is *correct*, then the probability of obtaining exactly such a combination of positive and negative feedbacks for the set of cycles F can be calculated as

$$l_c^+(F) \quad = \quad \prod_{f \in F^+} prob^+(\|f\|, \epsilon_{cyc}, \delta_{cyc}) \prod_{f \in F^-} (1 - prob^+(\|f\|, \epsilon_{cyc}, \delta_{cyc})).$$

This probability is the product of all individual probabilities for positive and negative feedback cycles of the given lengths, as the they have been previously derived in equations 1 and 2, to occur.

Similarly, if p_1 assumes that its own outgoing translation link at the start of the cycle is *incorrect*, then the probability of obtaining such a combination of feedbacks for the set F can be calculated as

$$l_c^-(F) \quad = \quad \prod_{f \in F^-} prob^-(\|f\|, \epsilon_{cyc}, \delta_{cyc}) \prod_{f \in F^+} (1 - prob^-(\|f\|, \epsilon_{cyc}, \delta_{cyc})).$$

Since we have no knowledge about ϵ_{cyc} and δ_{cyc} we assume these probabilities to be uniformly distributed. We integrate over ϵ_{cyc} and δ_{cyc} in order to obtain the expected probability for the distribution of positive and negative feedbacks in the observed set F to occur. We could take into account density functions here if we have any *a priori* knowledge about those two random variables. The resulting expectation values e_c^+ and e_c^- when assuming that the known translation $T_{p_1 \to p_2}$ is either correct or wrong, are then

$$e_c^+ = \int_0^1 \int_0^1 l_c^+(F) \, \mathrm{d}\epsilon_{cyc} \, \mathrm{d}\delta_{cyc}$$

$$e_c^- = \int_0^1 \int_0^1 l_c^-(F) \, \mathrm{d}\epsilon_{cyc} \, \mathrm{d}\delta_{cyc}$$

which are used to evaluate the relative degree of correctness γ_{cyc} of the mapping $T_{p_1 \to p_2}$ given the observation set F.

$$\gamma_{cyc} = \frac{e_c^+}{e_c^+ + e_c^-}.$$

If no relevant feedback is obtained for an attribute relative to a translation link we set by default $\gamma_{cyc} = 1$.

This analysis may be performed by any peer p_1 for every outgoing link to a peer p_j and every attribute $A_i \in as \cup ap$ independently, resulting in values $\gamma_{cyc,j}^{p_2}$ indicating the likelihood of the translation $T_{p_1 \to p_j}$ being correct for the attribute A_i.

As for the preceding section, we define now a feature vector and a similarity measure to capture the semantic losses along a sequence translation links $T_1, \ldots, T_n$, where T_j connects peer p_j with p_{j+1} via a translation link. For simplicity of presentation we assume each peer corresponds to a different semantic domain.

Let us suppose that peer p_1 issues a query $q = \pi_{ap}(\sigma_{pred(as)}(\mu_{fa}(DB)))$ to p_2 through a translation link $T_1 = T_{p_1 \to p_2}$. p_1 computes a feature vector for q based on the cycle messages it has received as follows:

$$\overrightarrow{FV^{\circlearrowleft}}(T_1(q)) = (FV_1^{\circlearrowleft}(T_1(q)), \ldots, FV_k^{\circlearrowleft}(T_1(q)))$$

where

$$FV_i^{\circlearrowleft}(T_1(q)) = \gamma_{cyc,i}^{p_2}.$$

In the following translations these values are updated by iteratively multiplying the values obtained for the degree of correctness for each translation link. We consider here that if two translations T_{j-1} and T_j have degrees of correctness of $\gamma_{cyc,i}^{p_j}$ and $\gamma_{cyc,i}^{p_{j+1}}$ for

attribute A_i and are independent, the degree of correctness of the composite translation $(T_{j-1} \circ T_j)$ is $\gamma_{cyc,i}^{p_j} \gamma_{cyc,i}^{p_{j+1}}$. Thus, when forwarding a transformed query using a link T_{j-1}, peer p_j updates each value $FV_i^{\circlearrowleft}((T_1 \circ \ldots \circ T_{j-1})(q))$ it has received along with the transformed query $(T_1 \circ \ldots \circ T_{j-1})(q)$ in this way:

$$FV_i^{\circlearrowleft}((T_1 \circ \ldots \circ T_j)(q)) = FV_i^{\circlearrowleft}((T_1 \circ \ldots \circ T_{j-1})(q)) \gamma_{cyc,i}^{p_{j+1}}.$$

The semantic similarity for transformations $T_1, \ldots, T_n$ associated with the vector $\overrightarrow{FV}$ is then

$$SIM_{\circlearrowleft}(q, (T_1 \circ \ldots \circ T_n)(q)) = \frac{\overrightarrow{W} \cdot \overrightarrow{FV^{\circlearrowleft}}}{\left|\overrightarrow{W}\right| \left|\overrightarrow{FV^{\circlearrowleft}}\right|}.$$

This 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.

We illustrate the cycle analysis by means of the example given in Fig. 5. Assume p_1 forwards query $q = \pi_{A_1,B_1,C_1}(DB_1)$ through translation links $T_{p_1 \to p_2}$ and $T_{p_2 \to p_1}$ and obtains as a result of this cycle f the positive feedback $source_{T_{p_1 \to p_2} \circ T_{p_2 \to p_1}}(A_1) = \{A_1\}$ for attribute A_1. It calculates $l_c^+(c) = prob^+(2, \epsilon_{cyc}, \delta_{cyc}) = (1 - \epsilon_{cyc}) + \epsilon_{cyc}\delta_{cyc}$. After integration it obtains a degree of correctness of $\gamma_{cyc,1}^{p_2} = \frac{3}{4}$. Since no feedback is obtained for the other attributes, p_1 sets $\overrightarrow{FV^{\circlearrowleft}}(T_{p_1 \to p_2}(q)) = (\frac{3}{4}, 1, 1)$, for the attributes occurring in q and calculates $SIM_{\circlearrowleft}(q, T_{p_1 \to p_2}(q)) \cong 0.957$. For the translation link $T_{p_1 \to p_3}$ to peer p_3 peer p_1 obtains feedback through translation links $T_{p_1 \to p_3}$ and $T_{p_3 \to p_1}$. For A_1 and B_1 this feedback is positive, whereas for C_1 it is negative. Doing the corresponding calculations this results in a feature vector $\overrightarrow{FV^{\circlearrowleft}}(T_{p_1 \to p_3}(q)) = (\frac{3}{4}, \frac{3}{4}, \frac{1}{4})$. p_1 calculates $SIM_{\circlearrowleft}(q, T_{p_1 \to p_3}(q)) \cong 0.763$.

When deciding to forward the query, assume that a peer requires all similarity measures (syntactic and semantic) to be above a threshold of 0.9 (see Section 6). Then it would not forward query q to peer p_2 for syntactic reasons (SIM_π is below the threshold), whereas it would not forward query q to p_3 for semantic reasons ($SIM_{\circlearrowleft}$ is below the threshold).

A more detailed example of cycle analysis is presented in Section 6.2.

5.2 Results Analysis

The second mechanism for analyzing the semantic quality of the translations is based on the analysis of the results returned. In [1] we have introduced a method using functional dependencies at the data level in order to assess the quality of translations. This method was based on analyzing to which extent integrity constraints are preserved after translation.

Here we present an alternative, more general, approach. We assume that peers annotate documents $\mathcal{D}$ using meta-data expressed according to our data model. Thus each document $d \in \mathcal{D}$ owned by peer p is associated with an annotation $annot(d)$ according to the schema S_p of the peer. Having sent a query, peers start to receive result documents with semantically rich content, e.g., images or full text. Based on this content they attempt to assess to which extent the queries expressed at the meta-data level were properly translated and thus led other peers to return the correct result documents.

Queries in our meta-data model are thus an intensional way of expressing semantic concepts, whereas extensionally the concepts are related to sets of documents. The problem that we address is of how to arrive at agreed annotation schemes at the intensional level that result in concept definitions that are compatible with the extensional notion of concepts that peers have.

In the following we assume that a peer has a finite set of concepts $\mathcal{C}$ to classify documents. The extensional notion of a concept that each peer has is based on methods of content analysis. Here, we do not make any assumption about the methods (e.g., layout

analysis, lexicographical analysis, contour-detection, etc., or even simple manual classification) used to extract meaningful features out of the documents; we simply treat them as high-level abstractions used to unambiguously classify any possible retrieved documents $d \in \mathcal{D}$ into concepts $c \in \mathcal{C}$ using a decision rule $\mathcal{R}_{content}$:

$$\mathcal{R}_{content} : \mathcal{D} \rightarrow \mathcal{C}.$$

In a more general setting, $\mathcal{R}_{content}$ could be a probabilistic rule. Using their local classification based on content analysis, peers can thus determine for every received document the concept it belongs to.

The intensional notion of concept each peer has is based on classification rules applied to metadata annotations of documents.

$$\mathcal{R}_{annot} : annot(\mathcal{D}) \rightarrow \mathcal{C}.$$

Again, we do not make assumptions on the specific form of the classification rules, except that they apply some predicates to the metadata annotations and derive from these predicates the concept to which the document corresponds to. Examples of classification rules are extensively discussed in the data mining literature. The document classification obtained from content analysis and by classification rules are presumed to be consistent up to a mean classification error ϵ_{res}, i.e., we assume that with a probability $1 - \epsilon_{res}$

$$\mathcal{R}_{content}(d) = \mathcal{R}_{annot}(annot(d)).$$

By analyzing its own document collection a peer can estimate the value of ϵ_{res}.

Imagine now a peer p_1 classifying documents according to rules $\mathcal{R}^{p_1}_{content}$ and $\mathcal{R}^{p_1}_{annot}$. Peer p_1 issues a query q against the metadata annotation for retrieving documents. Upon reception of a document d from a foreign peer $p_2 \in N_e(p_1)$, p_1 performs the classification operation according to its own rules $\mathcal{R}^{p_1}_{content}$ and $\mathcal{R}^{p_1}_{annot}$. Different situations may then occur:

- $\mathcal{R}^{p_1}_{content}(d) = \mathcal{R}^{p_1}_{annot}(d)$: this is the result p_1 was expecting; it is an indication that the outgoing translation link used to forward q to p_2 was semantically correct for query q. We treat this as positive feedback (F^+).
- $\mathcal{R}^{p_1}_{content}(d) \neq \mathcal{R}^{p_1}_{annot}(d)$: p_1 receives a document, such that the content analysis does not match the classification obtained from the metadata annotation obtained by translation. Since the document content is not changed during transmission of the query result, this implies that some semantic confusion occurred in the metadata query translation along the path from p_1 to p_2. In this case, we consider this as negative feedback (F^-).

If p_1 and p_2 are directly connected, this gives us a clear indication about the semantic (in)correctness of the translation link $T_{p_1 \rightarrow p_2}$. Given the mean classification error probability ϵ_{res}, the probability of the link being correct or incorrect in case of positive feedback are $1 - \epsilon_{res}$ and ϵ_{res} respectively. In case of negative feedback, they become ϵ_{res} and $1 - \epsilon_{res}$.

If two peers are separated by one or more semantic domains, the situation is somewhat more complicated since we have to take into account all the successive links used to forward the query from p_1 to a peer p_n. Let us suppose that a peer receives some feedback f after the query has gone through $\|f\|$ different translation links; analogous to the derivation of the probabilities from the cycle analysis, the probability of receiving a positive feedback assuming the link we are analyzing is correct is

$$(1 - \epsilon_{res}) prob^+(\|f\| - 1, \epsilon_{cyc}, \delta_{cyc}) + \epsilon_{res}\delta_{res}(1 - prob^+(\|f\| - 1, \epsilon_{cyc}, \delta_{cyc})),$$

where $prob^+$ is defined as in equation (1). The first term covers the case where the translations are all correct and the peer performs a proper classification, and thus obtains positive

feedback. The situation where the intermediate translations are wrong and the peer still believes to have obtained a positive feedback is more intricate and is covered by the second term. Receiving a wrongly annotated result a peer can still perform a misclassification itself with probability ϵ_{res}. However, only in exceptional cases with probability δ_{res} this misclassification will correct the problem, namely when the "wrong concept" matches exactly the expected concept. A peer can estimate the probability δ_{res} by $(\|\mathcal{C}\| - 1)^{-1}$, where $\|\mathcal{C}\|$ is the number of different concepts a peer knows at a given instant of time. The probability of receiving negative feedback is then calculated analogously.

Performing an analysis analogous to the one given in Section 5.1 and introducing l_r^+ and l_r^- as the probability of receiving a certain combination of responses for a given error model under the assumption that the outgoing translation link is correct resp. incorrect, we obtain again two expectation values e_r^+ and e_r^- used to estimate the degree of semantic correctness:

$$e_r^+ = \int_0^1 \int_0^1 l_r^+(F)\,\mathrm{d}\epsilon_{cyc}\,\mathrm{d}\delta_{cyc}$$

$$e_r^- = \int_0^1 \int_0^1 l_r^-(F)\,\mathrm{d}\epsilon_{cyc}\,\mathrm{d}\delta_{cyc}.$$

Defining $\gamma_{res}^{p_2} = \frac{e_r^+}{e_r^+ + e_r^-}$ as the likelihood of the translation $T_{p_1 \to p_2}$ being correct for a peer $p_2 \in N_e(p_1)$ we obtain a scalar feature for each translation link $T_{p_1 \to p_2}$

$$FV^{\rightleftarrows}(T_{p_1 \to p_2}(q)) = \gamma_{res}^{p_2}$$

measuring the degree of correctness of the translation link. If no value can be computed it is again set to 1 by default. Analogous to the cycle analysis these values are forwarded and updated iteratively by multiplying the values obtained for each translation link, such that a measure for the semantic similarity

$$SIM_{\rightleftarrows}(q, (T_1 \circ \ldots \circ T_n)(q)) = \gamma_{res}^{p_2} \ldots \gamma_{res}^{p_{n+1}}$$

for a chain of translations is defined.

Some illustrating examples for this approach are given in Section 7.

6 Gossiping Algorithm

6.1 Query Forwarding

At this point, we have four measures (SIM_σ, SIM_π, $SIM_{\circlearrowleft}$ and $SIM_{\rightleftarrows}$) for evaluating the losses due to the translations. We will now make use of these values to decide whether or not it is worth forwarding a specific query to a foreign semantic domain.

First, we require the creator of a query to attach a few user-defined or generated values to the query it issues:

- The weights $\overrightarrow{W}$ pondering the importance of the attributes in the query.
- The respective selectivity of the selection attributes $\overrightarrow{sel}$.
- The minimal values $\overrightarrow{SIM}_{min} = (SIM_\sigma^{min}, SIM_\pi^{min}, SIM_{\circlearrowleft}^{min}, SIM_{\rightleftarrows}^{min})$ for the similarity measures under which a transformed query is so deteriorated that it can no longer be considered as equivalent to the original query.

We extend the format of a query message to include these values as well as the iteratively updated feature vectors:

$$query(id, q, p, TT, \overrightarrow{W}, \overrightarrow{sel}, \overrightarrow{SIM_{min}}, \overrightarrow{FV_{\sigma}}, \overrightarrow{FV_{\pi}}, \overrightarrow{FV_{\circlearrowleft}}, FV_{\rightleftarrows}).$$

Now, upon reception of a query message, we require a peer to perform a series of tasks:

1. detect any semantic cycles
2. check whether or not this query has already been received
3. in case the local neighborhood has not received the query, forward it to the local neighborhood
4. return potential results

and, for each of its outgoing translation links:

5. apply the translation to the query
6. update the similarity measures for the transformed query
7. perform a test for each of the similarity measures whether the current similarity of the transformed query with the original query exceeds the required minimal threshold given by $\overrightarrow{SIM_{min}}$.
8. forward the query using the link if all similarity measure tests succeed.

This algorithm ensures that queries are forwarded to a sufficiently large set of peers capable of rendering meaningful feedback without flooding the entire network.

6.2 Case Study Revisited - Use of Syntactic and Semantic Similarities

Let us come back to the case study introduced in Section 3.3. We assume that a single attribute query is issued by p_1 to obtain all the titles of the different projects. This query may be written in the following way:

```
Query = FOR $project IN "project_A.xml"/* RETURN
<title>$project/title</title>
```

Let us now determine how the query will be propagated from p_1. Note that the weight and selectivity values attached to the query do not matter here, as a single attribute is concerned. Moreover we will not consider SIM_{σ} here (SIM_{σ} always evaluates to 1 because there is no selection attribute). The other thresholds are set to 0.5.

Following the gossiping algorithm, p_1 first attempts to transmit the query to its direct neighbors, i.e., p_2, p_3 and p_4. p_2 and p_4 in turn forward the query to the other nodes, but p_3 will in fact never receive the query: As p_3 has no representation for the *title*, the only projection attribute would be lost in the translation process from p_1 to p_3, lowering SIM_{π} to 0.

Let us now examine the semantic similarity $SIM_{\circlearrowleft}$. For the topology considered, thirty-one semantic cycles could be detected by p_1 in the best case. As the query never traverses p_3, only eight cycles remain (Table 1 lists those cycles). Now we use the formulas from Section 5: For its first outgoing link (i.e., the link going from p_1 to p_2), p_1 receives five positive cycles, raising the semantic similarity measure for this link and the attribute considered to 0.79.[2] p_1 does not receive any semantically significant feedback for its second outgoing link $T_{p_1 \to p_3}$, which is anyway handled by the syntactic analysis. Yet, it receives three negative cycles for its last outgoing link $T_{p_1 \to p_4}$. This link is clearly semantically erroneous, mapping *title* onto *acronym*. This results in p_1 excluding the link for forwarding the query, since the semantic similarity drops to 0.26 in this case.

[2] Remember that we did not make any assumption regarding the distribution of erroneous links. In this case, the positive feedback received may as well come from a series of compensating errors.

Cycle	$T_{p_1 \to p_4}$ erroneous	$T_{p_2 \to p_4}$ erroneous
p_1, p_2, p_4, p_5, p_1	+	-
$p_1, p_2, p_4, p_5, p_6, p_1$	+	-
p_1, p_2, p_5, p_1	+	+
p_1, p_2, p_5, p_6, p_1	+	+
p_1, p_2, p_6, p_1	+	+
p_1, p_4, p_5, p_1	-	+
$p_1, p_4, p_5, p_2, p_6, p_1$	-	+
p_1, p_4, p_5, p_6, p_1	-	+

Table 1: Cycles resulting in positive (+) or negative (-) feedback

The situation may be summarized in this way: p_1 restrains from sending the query through p_3 because of the syntactic analysis (too much information is lost in the translation process) and excludes p_2 because of the high semantic dissimilarity.

The situation somewhat changes if we correct the erroneous link $Tp_1 \rightarrow p_4$ and add a mistake for the link $Tp_2 \rightarrow p_4$. For the attribute considered, the semantic similarity drops to 0.69 for the outgoing link $Tp_1 \rightarrow p_2$ (two long cycles are negative, see third column in Table 1). Even though it is not directly connected to an erroneous link, p_1 senses the semantic incompatibilities affecting some of the messages traversing p_2. It will continue to send queries through this link, as long as it receives positive feedback at least.

7 Experimental evaluation

In the preceding section, we have evaluated the Chatty Web approach by examining query forwarding in a small network of static translations generated by a group of users. In contrast to this, we now use semantic gossiping and the semantic similarity measures not only to decide on query forwarding but also to correct existing mappings. Thus semantic gossiping is used to automatically reach semantic agreement in large networks of computer-generated and dynamic translation links. This approach in place could for example be used to derive basic, common ontologies from a dynamic system with heterogeneous schemas, or to gradually refine existing networks of translations. The initial simulation results interpreted below provide promising evidence that it is worth pursuing further research along these lines and highlight some of the issues to be addressed. In particular, they clearly indicate in which settings each of the two semantic similarity measures derived from cycle and result analysis are more suitable.

7.1 Experimental setup

The setup we used in the experiments is as follows: We assume a network of peers representing individual semantic domains. Peers share a finite set of similar concepts, i.e., operate in a certain semantic domain (for example, biological databases) inside the network. They share annotated documents (or data) related to those concepts, but refer to concepts using different names (they denominate the concepts differently). From this basic setup, we attempt to create global interoperability by applying semantic gossiping techniques using purely pair-wise, local translations.

The exact description of the process is as follows: First, we create a topology of n peers $p_1 \ldots p_n$, each of them connected through translation links to l other peers. The peers share $\|\mathcal{C}\|$ concepts $c_1 \ldots c_{\|\mathcal{C}\|}$, but use distinct names to refer to them. Thus we study the problem of peers sharing the same concepts but lacking knowledge of how to refer to them by names. This is somewhat similar to the approach taken in [29], without aiming at universally agreed upon names. Without loss of generality we may assume that the same set of names $n_1 \ldots n_{\|\mathcal{C}\|}$ is used by all peers (this simplifies the subsequent presentation). We write $(n_i \mapsto_p c_k)$ if peer p uses name n_i to refer to concept c_k. Thus, we can use a single attribute A to store the name the peer associates with a concept. Also, peers can verify whether a document belongs to a concept or not and thus annotate documents they store with a name using attribute A .

We then generate mappings $\mu(DB)$ for every translation link. The mapping function μ relates names from the first peer to names from the second peer, with every name used by the first peer mapped onto the name used by the second peer. Thus μ is a permutation of the domain of names used for attribute A which we denote as $\mu(n_i) = n_j$ to indicate that μ maps name n_i to name n_j. For every mapping $\mu_{p_1 \rightarrow p_2}$ in every translation link $T_{p_1 \rightarrow p_2}$, we say that the mapping is correct if and only if the two names bound by the mapping actually refer to the same concept, that is if

$$\mu_{p_1 \to p_2}(n_i) = n_j \wedge n_i \mapsto_{p1} c_k \wedge n_j \mapsto_{p2} c_k.$$

Thus, random mappings would only have a probability of $\frac{1}{\|C\|^!}$ of being correct in this setting. In the experiments, we generate a fraction $eRate$ of erroneous mapping initially.

Unless specified otherwise, we use small-world graphs [31] to interconnect peers with translation links since small-world topologies have been extensively applied to model computer networks or social behaviors. They are typically characterized by high clustering coefficients (average fraction of pairs of neighbors of a node that are also neighbors of each other) and relatively small path length (average minimal distance between two nodes). In the following, we generate graphs with an average clustering coefficient of 0.1 and with 10% shortcuts (i.e., links rewired to a random peer in the network).

Starting from the original topology, we apply semantic gossiping techniques iteratively in order to detect and rectify erroneous translations. At every simulation step, each peer selects one of its names randomly and issues a query about this name (i.e., the query consists of a projection on one attribute: the name selected). The query is propagated to the other peers (semantic domains) in a Gnutella-like fashion with a low time-to-live (TTL) value.

The syntactic analysis for this simplistic type of query is straightforward: peers forward the query through an outgoing translation link if there exists a translation mapping the local name used in the query (projection attribute) into another name for the foreign peer. Now, for detecting and repairing erroneous translation links, we slightly modify the semantic analysis; we forward queries irrespectively of the results of previous query forwarding strategy in order to get as many evidences as possible, and use these results to reach semantic agreements by gradually modifying translations.

Before taking a closer look at the final results, we will evaluate in the following sections each of the semantic analyses (cycle and result analysis) separately to emphasize their specificities.

7.2 Cycle Analysis

For every iteration step, peers randomly choose a name, send a query for this name and analyze the cycle messages they get in return. Here, we do not only estimate the correctness of the actual mapping as explained in Section 5.1, but also determine which of the possible mappings is most likely correct and adopt it as a new mapping. Therefore, peers view mappings resulting from returned queries as new mapping candidates. Consider for example Fig. 7, where peer p_1 systematically receives n_1 mapped onto n_2 in returned queries (negative feedback). In addition to evaluating the correctness of the current mapping, p_1 considers other mappings as well. It adopts the most probably correct mapping candidate if its probability of being correct is above 50%. In this example, p_1 evaluates the correctness of mapping n_1 onto n_2, and might consider to modify it to a mapping n_1 onto n_1.

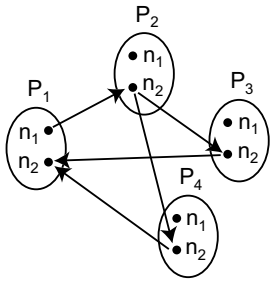

Figure 7: New mapping candidates

As indicated in Section 5.1, preexisting knowledge on the distribution of error probabilities δ_{cyc} and ϵ_{cyc} may be used in the computation of semantic similarity. δ_{cyc}, the

probability of a series of different errors to compensate along a cycle, is approximated to $(\|\mathcal{C}\| - 1)^{-1}$, which is the probability of the last erroneous link in the cycle to map to the original name and thus to correct previous errors.

We estimate ϵ_{cyc} with standard maximum-likelihood techniques applied to the feedback information we receive. From the probability of receiving a positive cycle of length $\|f\|$ knowing that the error probability of a translation link is ϵ_{cyc},

$$(1 - \epsilon_{cyc})^{\|f\|} + (1 - (1 - \epsilon_{cyc})^{\|f\|})\delta_{cyc},$$

and from its negative counterpart, we derive the density function for the likelihood of ϵ_{cyc}:

$$L(\epsilon_{cyc}|F) =$$

$$K \prod_{f^+ \in F^+} ((1-\epsilon_{cyc})^{\|f^+\|} + (1-(1-\epsilon_{cyc})^{\|f^+\|})\delta_{cyc}) \prod_{f^- \in F^-} (1-(1-\epsilon_{cyc})^{\|f^-\|})(1-\delta_{cyc})$$

where K is a normalizing constant. The local maximum of this function over $[0, 1]$ gives a good approximation of ϵ_{cyc}, supposing we have sufficient feedback information.

What is the result of this process in the long run? It depends of course on the initial setting but in the end, this method attempts to obtain a mapping consensus based on the different feedback cycles detected in 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. For settings where links are scarce, peers do not have sufficient information for making sensible choices, and results may diverge.

Several parameters are of particular interest: The number of peers n, the fraction of translations initially erroneous $eRate$, the number of concepts $\|\mathcal{C}\|$, the initial time-to-live TTL of the messages and the number of outgoing translation links l per peer. The figures below show experimental results for topologies where $n = 25$, $eRate = 0.1$, $\|\mathcal{C}\| = 4$, $TTL = 5$ and $l = 5$ and where one of those parameters varies. All the curves are averaged over ten consecutive runs. At every step, each peer sends a query picking a random concept for every outgoing edge and modifies its mappings depending on the results of the analysis explained above. Steps are represented on the x-axis. The graph shows the evolution of the percentage of erroneous mappings, starting at a rate $eRate$ initially. Clearly, the outcome depends on the density of links, which directly impacts on the number of cycles we have at our disposal for taking mapping decisions (see Fig. 8). For $l = 4$ and the topology considered, we get on average only one positive feedback per mapping candidate, which is obviously insufficient to take sensible decisions. For $l = 5$ and $l = 6$, the value raises to 1.8 and 2.9 respectively and most of the erroneous mappings get corrected after ten iterations. Finally, for $l = 7$, we get enough evidences (4.5 per mapping candidate on average) for correcting all the erroneous links, thus reaching a perfect semantic agreement, in eight steps.

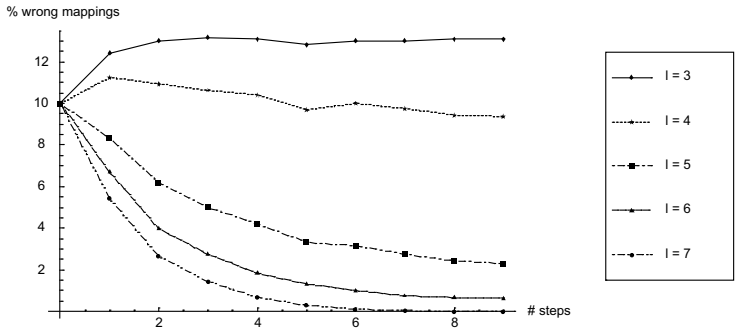

Figure 8: Sensitivity to the number of outgoing edges

Similar results may be observed for variable TTLs. Fig. 9 shows results using the same parameters as before, but this time for a fixed number of outgoing edges ($l = 4$) and TTLs ranging from 3 to 6. Again, for low values, peers do not gain sufficient feedback information to correct mappings. Starting with $TTL = 4$ (1.8 positive feedbacks per decision), peers receive sufficient information to correct more than 75% of the erroneous mappings after nine iterations. Low-connectivity networks may thus benefit from increasing the TTL value of their queries in order to get sufficient feedback information for the peers.

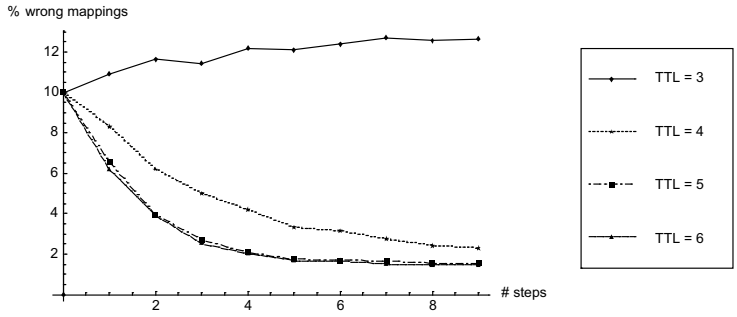

Figure 9: Sensitivity to the TTL

Our approach is rather insensitive to variations of the initial error rate (see Fig. 10) until a certain threshold, where too many bad links are present initially to reach a correct consensus based on the feedback cycles. Finally, it is worth mentioning that the approach scales very well with the number of nodes. This is not surprising, considering that the method relies solely on local interactions (no central component or computation) and that the clustering coefficient of the network is relatively high. Fig. 11 shows experiments for networks ranging from 50 to 800 peers without fundamental results variations. The small deviations are due to the *shortcuts* in the small world topology which connect two random peers in the network. The bigger the graph, the less likely it is that these links can be used to form cycles within a certain neighborhood.

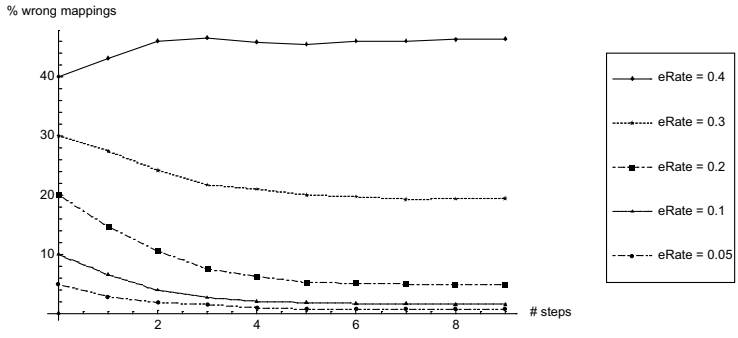

Figure 10: Sensitivity to the initial error rate

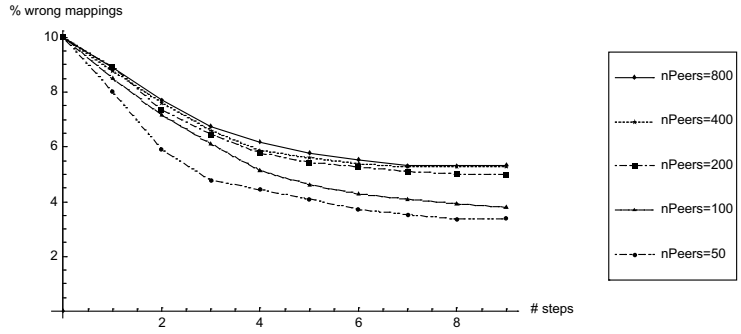

Figure 11: Scalability

7.3 Results Analysis

Let us now consider the second part of the analysis, in which peers analyze and categorize documents they receive. The process is as follows: At every step, peers first issue a couple of queries with a high TTL for estimating the error rate as explained in the preceding section. Then, for each of their outgoing links, the peers pick a concept randomly and issue a query asking for documents related to that concept. In return, they receive documents they analyze following the method described in Section 5.2. They modify the mapping they have used to forward the query with the most probable mapping if it has a correctness likelihood of at least 0.5.

For the simulations, we used a fixed set of documents scattered randomly among the peers. All documents are assigned to concepts. Each document owner has a probability (ϵ_{res}) of misclassifying a document by relating it to a wrong concept. We use a fixed, low value of $\epsilon_{res} = 5\%$ in the following experiments. For our setting, δ_{res} is equal to $(\|\mathcal{C}\| - 1)^{-1}$.

Unless specified otherwise, we used a network of 50 peers sharing in total 100 documents, 2 outgoing translation links per peer, 4 concepts, a TTL of 3, an initial error rate of 10%, and a probability of 10% of misclassifying documents.

First, it is interesting to see that this approach is very robust against the initial error rate, mainly because of the short feedback loop (one translation link suffices here to return documents) compared to the relatively long cycles used previously. Fig. 12 shows the results for a varying initial percentage of wrong mappings.

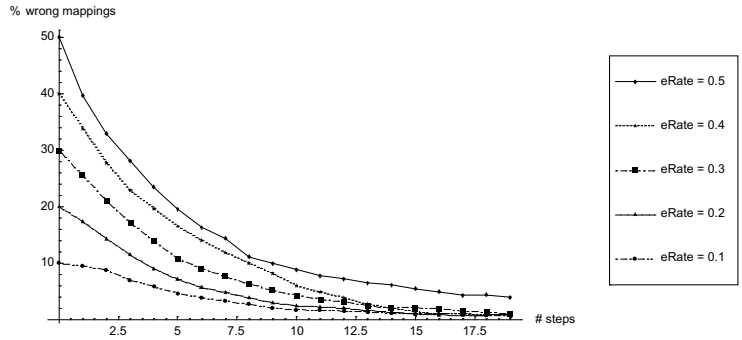

Figure 12: Sensitivity to initial error rate

Nevertheless, the approach is rather sensitive to the rate of misclassification of documents, as shown in Fig. 13. This is especially true since we do not try to evaluate this parameter but consider a mere fixed value.

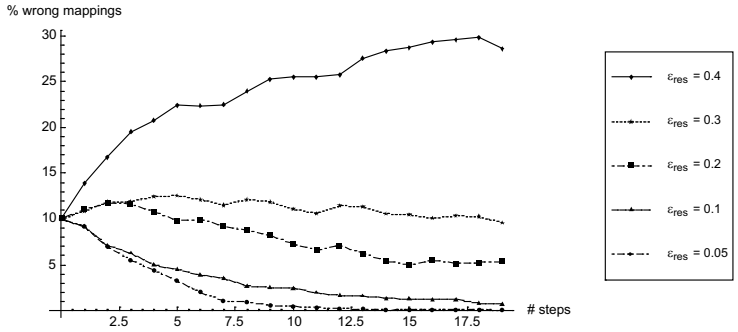

Figure 13: Sensitivity to misclassification rate

The approach taken here is completely local, and does not take into consideration any global behavior, and scales well with the number of peers (see Fig. 14). Here, we increase the number of documents linearly with the number of peers, to keep the average number of documents per peer constant. This number is essential to this analysis, since it is directly proportional to the number of evidences a peer gets for every query. This effect is depicted in Fig. 15: Peers start having trouble correcting the mappings as they get less and less documents returned for their queries (documents scarcity).

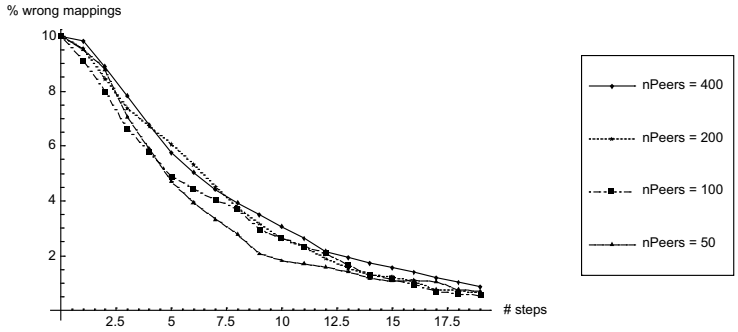

Figure 14: Scalability

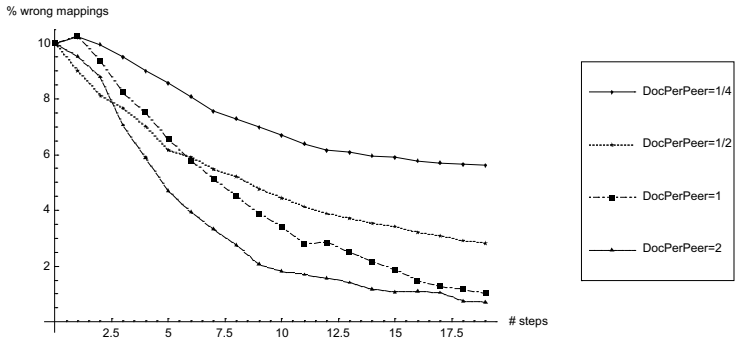

Figure 15: Sensitivity to number of documents

7.4 Combined Results

Many possibilities exist for combining the two analyses. We chose a very simple one: at each step, every peer first performs a result analysis step (modifying a few 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). The results for topologies with 25 peers, 4 concepts, 2 outgoing edges, TTLs of 3 (results) or 6 (cycles) and varying error rates on initial mappings are depicted in Fig. 16.

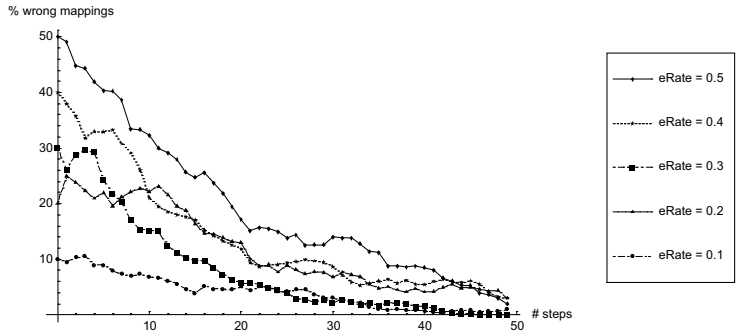

Figure 16: Combined results, varying initial error rate

This method takes more time 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 analyses point of view. Note that the combined method in the end outperforms the two individual methods applied separately (e.g., more than 95% of erroneous mappings corrected after 50 steps with 50% erroneous mappings initially).

8 Implementation framework

All the tasks of the Chatty Web approach have been been mapped onto an implementation architecture which uses a meta-data model expressed in XML and XQuery as the language to translate among schemas. The framework assumes the availability of a communication infrastructure, for example, simple web access via HTTP or a P2P infrastructure such as JXTA [9]. However, we are not bound to any specific communication infrastructure. All we require is access to the relevant schema data and the ability to query information and results. This can easily be achieved by a standard abstraction layer that maps a specific communication infrastructure's interface to the one we require. Since this is a fairly standard software engineering task we omit it in the following discussion. Based on these assumptions, Fig. 17 shows the standard architecture used for semantic gossiping in the Chatty Web.

Incoming queries are registered at and handled by the *Incoming Query and Result Handler* whose task is to communicate with other peers, to forward the query for further processing and to gather partial results which it uses to assemble the final result of a specific query. The next step then is to detect whether a cycle has occurred. If so, semantic analysis of the cycle is triggered. Otherwise, the query is processed, first by querying the local database and then by handing it over to the *Query Router and Translator* to collect results from other peers.

For this purpose the *Query Router and Translator* inquires for possible translations, evaluates the quality of the resulting queries, and if it is above a defined threshold, forwards the query to the respective peer in a different semantic domain. Queries are forwarded by

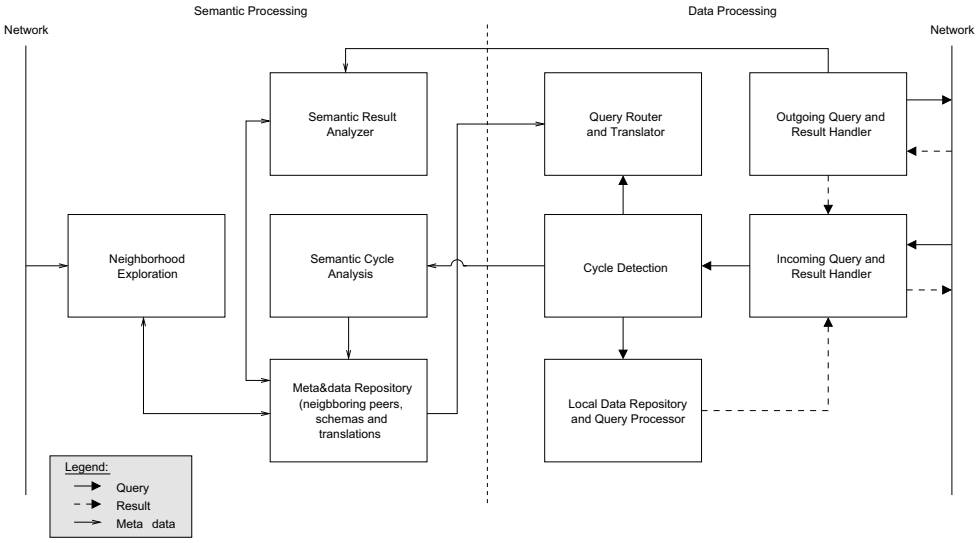

Figure 17: Architecture for semantic gossiping

the *Outgoing Query and Result Handler* which is also in charge of collecting the results and forwarding the results to the *Incoming Query and Result Handler* which returns them to the original requester. Additionally, it provides input data for semantic result analysis.

This is the main data processing flow of the architecture. In parallel, partly triggered by the ongoing data processing, there is also semantic processing as depicted in the left half of Fig 17. Its main tasks are semantic analyses of results based on the existing knowledge of schemas and their relationships and the semantic analyses of detected cycles. The results of these analyses are integrated again into the system's knowledge base and provide the basic decision criteria for query routing.

Additionally, the knowledge base is updated and improved by exploring the peer's neighborhood and detecting new schemas and translations. The meta-data repository will try to infer further translations and present new ones for human analysis or apply them for actively detecting semantic agreements in an automatic way.

9 Related Work

A number of approaches for making heterogeneous information sources interoperable are based on mappings between distributed schemas or ontologies without making the canonical assumption on the existence of a global schema.

For example, in OBSERVER [17] each information source maintains an ontology, expressed in description logics, to associate semantics with the information stored and to process distributed queries. In query processing, OBSERVER uses local measures for the loss of information when propagating queries and receiving results. Similarly to OBSERVER, KRAFT [25] proposes an agent-based architecture to manage ontological relationships in a distributed information system. Relationships among ontologies are expressed in a constraint language. [2] proposes a model and architecture for managing distributed relational databases in a P2P environment. The authors use local relational database schemas and represent the relations between those with domain relations and coordination formulas. These are used to propagate queries and updates. The relationships given between the local database schemas are always considered as being correct. In [24] a probabilistic framework for reasoning with assertions on schema relationships is introduced. Thus the

approach deals with the problem of having possibly contradictory knowledge on schema relationships. [20] proposes an architecture for the use of XML-based annotations in P2P systems to establish semantic interoperability.

An approach to self-organizing vocabularies is described in [29]. A set of agents communicate by randomly associating a fixed set of words to a fixed set of meanings (which is called a vocabulary but in fact is an ontology) and repeatedly evaluate how successful their communicative acts have been. Depending on the success, the binding between a word and a concept is maintained or replaced by a new random coupling. The decision is based on sigmoid functions so that the probability of change quickly decreases if the majority of of agents uses the same coupling. This approach is related to the method of cycle analysis we use and simulate in Section 7. However, it does not employ result analysis. Nevertheless [29] shows that semantic agreements are reached rather quickly. The additional result analysis we perform may help to speed up convergence speed and increase the scalability and robustness of the self-organization process. It is interesting to note that [29] shows that an increased numbers of agents, words, and meanings does not lead to combinatorial explosion but implosion. This is due to the fact that the increasing number of words with consistent meaning narrows the selection space drastically. This phenomenon is similar to the combinatorial implosions described by Kauffman [13] for the clustering and interconnection of autocatalytic networks.

Edutella [21] is a recent approach to apply the P2P architectural principle to build a semantically interoperable information system for the educational domain. The P2P principle is applied at the technical implementation level whereas logically a commonly shared ontology is used. The original design of Edutella which is based on Gnutella is changed to a super-peer network approach in [22] which offers better scalability and provides sophisticated routing and clustering strategies based on the meta-data schemas attributes and ontologies used. This approach includes a methodology for mediation between local schemas at super peers which enables super-peers to route queries and combine results from different semantic domains into one result. It employs transformation rules, so-called correspondences, which have already been used in mediator-based information systems [32]. *Query Response Assertions* [16] and *Model Correspondences* [3] are used to express correspondences between heterogeneous schemas.

The Piazza system [10] defines a mapping language to specify mappings between sets of XML or RDF data sources that tries to take into account both domain and document structure in the mediation process. The transitive closure of these mappings is used to provide a query answering algorithm over the graph of data source defined by the mappings. Piazza's approach is complementary to our approach since it assumes the existence of pairwise mappings between data sources and uses these mappings for answering queries while we try to detect the quality of mappings in terms of an overall agreement among nodes (which can also be seen as a form of transitive closure). However, the mapping language of Piazza together with its query rewriting and query answering methods could also be used in the Chatty Web approach for more expressive mappings and improved query routing.

Approaches for automatic schema matching—see [27] for an overview—would ideally support the approach we pursue in order to generate mappings in a semi-automated manner. In fact, we may understand our proposal as extending approaches for matching two schemas to an approach matching multiple schemas in a networked environment. One example illustrating how the schema matching process could be further automated at the local level is introduced in GLUE [6] which employs machine learning techniques to assist in the ontology mapping process. GLUE is based on a probabilistic model, employs similarity measures and uses a set of learning strategies to exploit ontologies in multiple ways to improve the resulting mappings.

Finally, we see our proposal also as an application of principles used in Web link analysis, such as [14], in which local relationships of information sources are exploited to derive global assessments on their quality (and eventually their meaning).

10 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. The success of the Semantic Web, however, depends heavily on the degree of global agreement that can be achieved, i.e., global semantics. In this paper we have presented an approach facilitating the fulfillment of this requirement by deriving global semantics (agreements) from purely local interactions/agreements. This means that explicit local mappings are used to derive an implicit global agreement. We see our approach as a complementary effort to the on-going standardization 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. To demonstrate its validity and practical usability, the model is applied in a simple yet practically relevant case study. Also, series of experimental results legitimate our claims and illustrate our interests in pursuing research aiming at a better understanding of network-related properties fostering semantic interoperability.

References

[1] K. Aberer, P. Cudr´e-Mauroux, and M. Hauswirth. The Chatty Web: Emergent Semantics Through Gossiping. In *International World Wide Web Conference (WWW)*, 2003.

[2] P. A. Bernstein, F. Giunchiglia, A. Kementsietsidis, J. Mylopoulos, L. Serafini, and I. Zaihrayeu. Data management for peer-to-peer computing: A vision. In *Workshop on the Web and Databases (WebDB)*, 2002.

[3] S. Busse. *Model Correspondences in Continuous Engineering of MBIS*. PhD thesis, Logos Verlag, 2002.

[4] Clip2. The Gnutella Protocol Specification v0.4 (Document Revision 1.2), Jun. 2001. http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf.

[5] M. Dean, D. Connolly, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein. OWL Web Ontology Language 1.0 Reference, 2002. W3C Working Draft 29 July 2002. http://www.w3c.org/TR/owl-ref/.

[6] A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Learning to map between Ontologies on the Semantic Web. In *International World Wide Web Conference (WWW)*, 2002.

[7] D. Fensel, I. Horrocks, F. van Harmelen, S. Decker, M. Erdmann, and M. C. A. Klein. OIL in a Nutshell. In *International Conference on Knowledge Engineering and Knowledge Management (EKAW)*, pages 1–16, 2000.

[8] FhG-IPSI. IPSI-XQ - The XQuery Demonstrator, 2002.

[9] L. Gong. JXTA: A Network Programming Environment. *IEEE Internet Computing*, 5(3):88–95, May/June 2001.

[10] A. Y. Halevy, Z. G. Ives, P. Mork, and I. Tatarinov. Piazza: Data Management Infrastructure for Semantic Web Applications. In *International World Wide Web Conference (WWW)*, 2003.

[11] I. Horrocks. DAML+OIL: a Description Logic for the Semantic Web. *IEEE Data Engineering Bulletin*, 25(1):4–9, 2002.

[12] R. Hull. Managing Semantic Heterogeneity in Databases: A Theoretical Perspective. In *Symposium on Principles of Database Systems (PODS)*, pages 51–61, 1997.

[13] S. A. Kauffman. *The Origins of Order - Self-Organization and Selection in Evolution*. Oxford Univ. Press, 1993.

[14] J. M. Kleinberg. Hubs, authorities, and communities. *ACM Computing Surveys*, 31(4es), 1999.

[15] M. Koubarakis, C. Tryfonopoulos, P. Raftopoulou, and T. Koutris. Data Models and Languages for Agent-Based Textual Information Dissemination. In *International Workshop on Cooperative Information Agents CIA*, pages 179–193, 2002.

[16] U. Leser. *Query Planning in Mediator Based Information Systems*. PhD thesis, TU Berlin, 2002.

[17] 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. *Distributed and Parallel Databases*, 8(2):223–271, 2000.

[18] G. A. Modica, A. Gal, and H. M. Jamil. The Use of Machine-Generated Ontologies in Dynamic Information Seeking. In *International Conference on Cooperative Information Systems (CoopIS)*, pages 433–448, 2001.

[19] MPEG-7. Multimedia Content Description Interface, 1996. http://ipsi.fraunhofer.de/delite/Projects/MPEG7/.

[20] A. Mukherjee, B. Esfandiari, and N. Arthorne. A Peer-to-peer System for Description and Discovery of Resource-sharing Communities. In *IEEE Workshop on Resource Sharing in Massively Distributed Systems (RESH)*, 2002.

[21] W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palm´er, and T. Risch. EDUTELLA: a P2P networking infrastructure based on RDF. In *International World Wide Web Conference (WWW)*, pages 604–615, 2000.

[22] W. Nejdl, M. Wolpers, W. Siberski, C. Schmitz, M. Schlosser, I. Brunkhorst, and A. Löser. Super-Peer-Based Routing and Clustering Strategies for RDF-Based Peer-To-Peer Networks. In *International World Wide Web Conference (WWW)*, 2003.

[23] B. Omelayenko. Integrating Vocabularies: Discovering and Representing Vocabulary Maps. In *International Semantic Web Conference*, pages 206–220, 2001.

[24] A. M. Ouksel and I. Ahmed. Ontologies are not the Panacea in Data Integration: A Flexible Coordinator to Mediate Context Construction. *Distributed and Parallel Databases*, 7(1):7–35, 1999.

[25] A. D. Preece, K. Hui, W.A. Gray, Trevor J. M. Bench-Capon P. Marti, Zhan Cui, and Dean Jones. Kraft: An Agent Architecture for Knowledge Fusion. *International Journal of Cooperative Information Systems (IJCIS)*, 10(1–2):171–195, 2001.

[26] Apache XML Project. Xerces Parser, 2002.

[27] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal*, 10(4):334–350, 2001.

[28] M. Sintek and S. Decker. TRIPLE - A Query, Inference, and Transformation Language for the Semantic Web. In *International Semantic Web Conference*, pages 364–378, 2002.

[29] L. Steels. Self-organising vocabularies. In *Artificial Life V*, 1996.

[30] Y. Sure, M. Erdmann, J. Angele, S. Staab, R. Studer, and D. Wenke. OntoEdit: Collaborative Ontology Development for the Semantic Web. In *International Semantic Web Conference*, pages 221–235, 2002.

[31] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

[32] G. Wiederhold. Mediators in the Architecture of Future Information Systems. *IEEE Computer*, 25(3):38–39, 1992.

Karl Aberer is a full professor at EPFL since September 2000. There he is heading the Distributed Information Systems Laboratory of the School of Computer and Communications Sciences. His main research interests are on distributed information management, P2P computing, semantic web and the self-organization of information systems. He received his Ph.D. in mathematics in 1991 from the ETH Zurich. From 1991 to 1992 he was postdoctoral fellow at the International Computer Science Institute (ICSI) at the University of California, Berkeley. In 1992 he joined the Integrated Publication and Information Systems institute (IPSI) of GMD in Germany, where he became manager of the research division Open Adaptive Information Management Systems in 1996. He has published more than 80 papers on data management on the WWW, database interoperability and query processing, workflow systems and P2P data management. Recently he has been PC-Chair of DBISP2P 2003, RIDE 2001, DS-9, and ODBASE 2002. He is associate editor of SIGMOD RECORD and member of the editorial board of the VLDB Journal and Web Intelligence and Agent Systems.

Philippe Cudré-Mauroux is a research assistant from the Distributed Information Systems Laboratory at the Swiss Federal Institute of Technology in Lausanne (EPFL). He holds a B.S. in Communication Systems from EPFL, a M.S. in Multimedia Communications from Eurecom Institute (France), and a graduate degree in Distributed Systems from University of Sophia Antipolis (France). Prior to joining EPFL, he worked on Content Delivery Architectures for HP and IBM Watson Research. Philippe is currently pursuing a Ph.D. under the supervision of Prof. Karl Aberer. His research interests include peer-to-peer systems, auto-organizational networks and semantics in decentralized systems. He is a member of the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS).

Manfred Hauswirth is a senior researcher at the Distributed Information Systems Laboratory at the EPFL Lausanne since January 2002. He holds an M.S. (1994) and a Ph.D. (1999) in computer science from the Technical University of Vienna. Prior to his work at EPFL he was an assistant professor at the Distributed Systems Group at the TU Vienna where he still lectures courses on distributed systems. His research interests include peer-to-peer systems, e-commerce, push systems, event-based systems, world-wide web, and programming languages and he has published numerous papers in these areas and written a book on distributed software architectures. He was the principal researcher in the Minstrel push system project and a senior researcher in the OPELIX and MOTION EU projects and has project management and consulting experience with several research and industry projects. He is member of program committees of international scientific conferences and a member of IEEE and ACM.