The Chatty Web: Emergent Semantics Through Gossiping[*]

Karl Aberer, Philippe Cudré-Mauroux, Manfred Hauswirth
École Polytechnique Fédérale de Lausanne (EPFL)
CH-1015 Lausanne, Switzerland
[Karl.Aberer, Philippe.Cudre-Mauroux, Manfred.Hauswirth]@epfl.ch

ABSTRACT

This paper describes a novel approach for obtaining semantic interoperability among data sources 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 agreement in an evolutionary and completely decentralized process that solely relies on pair-wise, local interactions: Participants provide translations between schemas they are interested in and can learn about other translations by routing queries (gossiping). To support the participants in assessing the semantic quality of the achieved agreements we develop a formal framework that takes into account both syntactic and semantic criteria. The assessment process is incremental and the quality ratings are adjusted along with the operation of the system. Ultimately, this process results in global agreement, i.e., the semantics that all participants understand. We discuss strategies to efficiently find translations and provide results from a case study to justify our claims. Our approach applies to any system which provides a communication infrastructure (existing websites or databases, decentralized systems, P2P systems) and offers the opportunity to study semantic interoperability as a global phenomenon in a network of information sharing parties.

Categories and Subject Descriptors

H.2.5 [**Information Systems**]: Database Management—*Heterogeneous Databases, Data Translation*

General Terms

Algorithms, Management, Theory

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, like P2P systems. Despite a large number of approaches and concepts, such as federated databases, the mediator concept [17], or ontology-based information integration approaches [8, 13], 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 effort 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 [5], OWL [3] or DAML+OIL [7], and based on shared ontologies. Still, these approaches rely on commonly agreed upon ontologies, which existing information sources can be related to by proper annotation. This is an extremely important development, but its success will heavily rely on the wide adoption of common ontologies or schemas.

The advent of P2P systems, however, introduces a dif-

[*]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.

WWW2003, May 20–24, 2003, Budapest, Hungary.
ACM 1-58113-680-3/03/0005.

ferent 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 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 or JXTA.

The first thing to observe is that semantic interoperability is always based on some form of agreement. Ontology-oriented approaches in the Semantic Web represent this agreement essentially *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 mappings 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 mappings even locally. However, a rich set of tools is getting available to to support this [16]. Establishing local agreements is a less challenging task than establishing global agreements by means of globally agreed 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.

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 such a gossiping-based approach depends clearly on the quality of the local mappings in the mapping graph. *Our fundamental assumptions is that these mappings may be incorrect.* Thus our agreement construction mechanisms try to determine which mappings 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 the different methods that can be applied to establish global forms of agreement starting from a graph of local mappings among schemas. We elaborate the details of each of these methods for a simple data model, that is yet expressive enough to cover many practical cases. Three methods will be introduced in particular:

1. A syntactic analysis of search queries after mappings have been applied in order to determine the potential information-loss incurred through the transformation.
2. A semantic analysis of composite mappings along cycles in the mapping graph, in order to determine the level of agreement that peers achieve throughout the cycle.
3. A semantic analysis of search results obtained through composite mappings based on the preservation of data dependencies.

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). We will provide results from initial experiments that have been performed for this setting.

We believe that this radically new approach to semantic interoperability shifts the attention from problems that are inherently difficult to solve in an automated manner at the global level ("How do humans interpret information models in terms of real world concepts?"), to a problem that leaves vast opportunities for automated processing and for increasing the value of existing information sources, namely the processing of existing local semantic relationships in order to raise the level of their use from local to global semantic interoperability. 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 mappings 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 mapping tools provide the mappings. The direction of the mapping and the node providing a mapping are not necessarily correlated. For instance, nodes A and B might both provide a mapping from $schema(A)$ to $schema(B)$, and they may exchange this mapping upon discretion. During the life-time of the system, each peer has the possibility to learn about existing mappings and add new ones. This means that a directed graph of mappings as shown in Figure 1 will be built between the neighborhoods along with the normal operation of the system (e.g., query processing and forwarding in a P2P system).

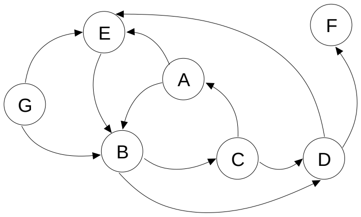


Figure 1: Mapping graph among semantic neighborhoods

This mapping graph has two interesting properties: (1) based on the already existing mappings and the ability to learn about existing mappings, new mappings can be added automatically by means of transitivity, for example, $D \rightarrow E \rightarrow B \Rightarrow D \rightarrow B$ and (2) the graph has cycles. (1) means that we can propagate queries towards nodes for which no direct translation link exists. This is what we call *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 are introduced below; (2) upon receiving results or feedback (cycle), it has to analyze the quality of the results at the schema and at the data level and adjust its criteria accordingly; and (3) update its view of the overall semantic agreement.

The criteria to assess the quality of translations—which in turn is a measure of the semantic agreement—can be categorized as *intrinsic* and *extrinsic*. Intrinsic criteria, discussed in Section 4, relate only to the processed query and to the required translation. We introduce the notion of *syntactic similarity* to analyze the extent to what a query is preserved after translation.

Extrinsic 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 (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 cycles. In case a disagreement is detected (e.g., a wrong attribute mapping at the schema level or the violation of a constraint at the data level), the peer has to suspect that at least some of the mappings involved in the cycle were incorrect, including the mapping it has used itself to propagate the query. Even if an agreement is detected, it is not clear whether this is not accidentally the result of compensating mapping errors along the cycle. Thus, analyses are required that assess which are the most probable sources of errors along cycles, to what extent the own mapping can be trusted and therefore of how to use these mappings in future routing decisions. At a global level, we can view the problem as follows: The translations among domains of semantic homogeneity (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 on an attribute-basis to the transformed queries and results in a *feature vector* (i.e., a vector encompassing the outcome of the criterion for each of the attributes concerned). The decision whether or not to forward a query using a translation link is based on 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 and where the correct mappings are increasingly reinforced by adapting the per-hop forwarding behaviors of the peers. Implicitly, this is a state where a global agreement on the semantics of the different schemas has been reached. This is illustrated by a case study 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 $t_1, \ldots, t_n$ 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 scientific databases, use exactly the type of simplistic data model we have introduced, at least at the meta-data level.

We assume that integrity constraints are known for the schema table R, both in terms of key constraints (uniqueness) and functional dependencies among attributes. Peers can answer queries q against their database. The result of a query q is denoted as $q(DB_p)$.

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 $\sigma_{p(a)}(R)$, where $a =< A_1, \ldots, A_k >$ is a list of attribute names, and p is any predicate on the attributes a using standard atomic predicates on the respective datatypes, i.e., $p = p(A_1, \ldots, A_k)$.
- 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 [2]. 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 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 Figure 2.

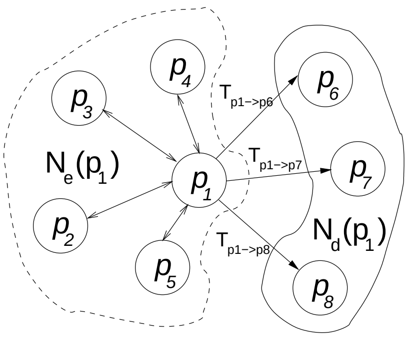


Figure 2: The Network Model

A peer p includes another peer p' with a different schema into its neighborhood if it knows a translation for queries against its own schema to queries against the foreign schema. The query translation operator $T_{p \rightarrow p'}$ is given by a query q_T that takes data structured according to schema $S_{p'}$ and transforms it into data structured according to schema S_p.

Thus $T_{p \rightarrow p'}$ has the property

$$T_{p \rightarrow p'}(q_p)(DB_{p'}) = q_p(q_T(DB_{p'}))$$

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

$$q_T(DB_{p'}) = \pi_a(\mu_f(DB_{p'}))$$

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. Note that multiple transformations may be applied to a single query iteratively:

$$T_{n-1 \rightarrow n}(\ldots T_{1 \rightarrow 2}(q) \ldots) = T_{1 \rightarrow 2, \ldots, n-1 \rightarrow n}(q)$$

Such query translations may be implemented easily using various mechanisms, for example XQuery, as done in our case study in Section 7.

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.

4. SYNTACTIC SIMILARITY

As intrinsic criterion to measure the degree of similarity between two queries (in our context, between an original query and a transformed query), we introduce the notion of *syntactic similarity.* Note that a high syntactic similarity (in terms of attributes lost during translation) will not ensure that forwarding the query is useful, but conversely a low syntactic similarity implies that it might not be useful to forward the query.

Let us suppose we have a query q, which always has the generic form of a selection-projection-mapping query

$$q = \pi_{ap}(\sigma_{p(as)}(\mu_{fa}(DB)))$$

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

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. Again, 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 to simplify the subsequent analysis.

Therefore the transformed query will be of the form (this is also true for multiple transformations after normalization)

$$T(q)(DB') = \pi_{ap}(\sigma_{p(as)}(\mu_{fa}(\pi_a(\mu_f(DB')))))$$

It might occur that attributes used in q are no longer available after the transformation T has been applied to q. This can only happen when an attribute needed for the derivation of a new attribute by means of one of the functions in fa and required in ap or as is missing, i.e., not occurring in a.

We now analyze which attributes are exactly needed in order to properly evaluate the query q. We define

$$att_\sigma(q) = \\ \{[A_0 : \{A_1, \ldots, A_k\}] \mid A_0 \in as, A_0 := F(A_1, \ldots, A_k) \in fa\}$$

and similarly

$$att_\pi(q) = \\ \{[A_0 : \{A_1, \ldots, A_k\}] \mid A_0 \in ap, A_0 := F(A_1, \ldots, A_k) \in fa\}$$

Given a transformation T we can define the source of an attribute $source_T(A)$:

If

$\exists F \in fa$ such that $A := F(A_1, \ldots, A_k)$

then

$source_T(A) = \{A_1, \ldots, A_k\}$

else

$source_T(A) = \perp$.

Informally, $source_T(A)$ tells whether and how an attribute is preserved in a transformation T. Then we can define the operation $\omega_T(att_\sigma(q))$ as follows:

If

$\forall [A_0 : \{A_1, \ldots, A_k\}] \in att_\sigma(q)$
$\forall A \in \{A_1, \ldots, A_k\} source_T(A) \neq \perp$

then

$[A_0 : \bigcup_{A \in \{A_1, \ldots, A_k\}} source_T(A)] \in \omega_T(att_\sigma(q))$

else

$[A_0 : \perp] \in \omega_T(att_\sigma(q))$.

This definition extends naturally to multiple transformation. In order to define

$$\omega_{T_n}(\ldots(\omega_{T_1}(att_\sigma(q)))\ldots)$$

we simply apply the above definition for ω_{T_n} to $\omega_{T_{n-1}}(\ldots(\omega_{T_1}(att_\sigma(q)))\ldots)$ instead of $att_\sigma(q)$. All definitions are analogous for $\omega_T(att_\pi(q))$.

$\omega_{T_n}(\ldots(\omega_{T_1}(att_\sigma(q)))\ldots)$ allows to determine which of the required attributes for evaluating queries are at disposal after applying the transformations $T_1, \ldots, T_n$. The definitions are given such that they 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 the transformed query and the original query.

The decision on the importance of attributes is query dependent. We have two issues to consider:

1. Not all attributes in as are preserved. Therefore some of the atomic predicates in $p(as)$ will not be correctly evaluated (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 $fv^\sigma_{A_i}$ for every attribute $A_i \in as \cup ap$ as follows:

   If

   $A_i \in as,\ [A_i : \perp] \in \omega_{T_n}(\ldots(\omega_{T_1}(att_\sigma(q))\ldots)$

   then

   $fv^\sigma_{A_i}(T_{1\rightarrow,\ldots,\rightarrow n}(q)) = 0$

   else

   $fv^\sigma_{A_i}(T_{1\rightarrow,\ldots,\rightarrow n}(q)) = sel_{A_i}$

   where sel_{A_i} is the selectivity (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) of an attribute A_i. Given the values $fv^\sigma_{A_i}$ for $A_i \in as \cup ap$ we can introduce a feature vector $\overrightarrow{FV_\sigma}$ for the transformed query $T_n(\ldots T_1(q)\ldots)$ characterizing the syntactic similarity with respect to the selection operator:

   $$\overrightarrow{FV_\sigma}(T_{1\rightarrow,\ldots,\rightarrow n}(q)) = (fv^\sigma_{A1}, \ldots, fv^\sigma_{Ak})$$

   We derive the syntactic similarity between the original query and the transformed query for the selection from this feature vector and from a user-defined weight vector $\overrightarrow{W} = (w_{A1}, \ldots, w_{Ak})$ with $A_i \in as \cup ap$ pondering the importance of the attributes:

   $$S_\sigma(q, T_{1\rightarrow,\ldots,\rightarrow n}(q)) = \frac{\overrightarrow{W} \cdot \overrightarrow{FV_\sigma}}{\left|\overrightarrow{W}\right| \left|\overrightarrow{FV_\sigma}\right|}$$

   where

   $$\overrightarrow{W} \cdot \overrightarrow{FV_\sigma} = w_1 fv^\sigma_{A1} + w_2 fv^\sigma_{A2} + \ldots + w_k fv^\sigma_{Ak}$$

   and where

   $$\left|\overrightarrow{X}\right| = \left\|\overrightarrow{X}\right\|_2 = \sqrt{x_1^2 + x_2^2 + \ldots + x_k^2}.$$

   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.

2. Not all attributes in ap are found in a or af. 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 propose to measure this for every attribute:

   If

   $$A_i \in as, \ [A_i : \perp] \in \omega_{T_n}(\ldots(\omega_{T_1}(att_\pi(q))\ldots)$$

   then

   $$fv^\pi_{A_i}(T_{1\rightarrow,\ldots,\rightarrow n}(q)) = 0$$

   else

   $$fv^\pi_{A_i}(T_{1\rightarrow,\ldots,\rightarrow n}(q)) = 1.$$

   The feature vector and the syntactic similarity for the projection operator then are

   $$\overrightarrow{FV_\pi}(T_{1\rightarrow\ldots\rightarrow n}(q)) = (fv^\pi_{A1}, \ldots, fv^\pi_{Ak})$$

   and

   $$S_\pi(q, T_{1\rightarrow,\ldots,\rightarrow n}(q)) = \frac{\overrightarrow{W} \cdot \overrightarrow{FV_\pi}}{\left|\overrightarrow{W}\right| \left|\overrightarrow{FV_\pi}\right|}.$$

   Again, this similarity decreases with the number of translations applied to the query, until it reaches 0 when all the projection attributes are lost.

5. SEMANTIC SIMILARITY

The intrinsic measure for syntactic similarity is based on the assumption that the query translations 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 extrinsic measures for 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 for the exact algorithm). Such a cycle starts with a peer p_1 transmitting a query q_1 to a second peer p_2 through a translation link $T_{1\rightarrow 2}$ (see Figure 3).

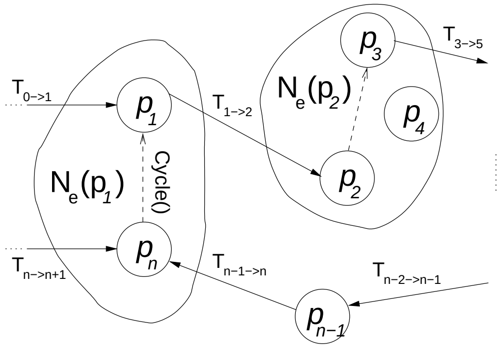


Figure 3: The Feedback Mechanism

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

$$q_n = T_{1\rightarrow 2, 3\rightarrow 5, \ldots, n-1\rightarrow n}(q_1)$$

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

– Case 1: $source_{T_{1\rightarrow\ldots\rightarrow n}}(A_i) = \{A_i\}$, 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 mapping.

– Case 2: $source_{T_{1\rightarrow\ldots\rightarrow n}}(A_i) = \perp$, 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 mapping unchanged.

– Case 3: Otherwise, if none of the two previous cases occurs, e.g., $source_{T_{1\rightarrow\ldots\rightarrow n}}(A_i) = \{A_j\}, j \neq i$, this indicates some semantic confusion along the cycle. Subcases can occur depending on what happens to A_j. This lowers the confidence in the mapping.

We then derive heuristics for p_1 to assess the correctness of the translation $T_{1\rightarrow 2}$ it has used based on the different cycle messages it received. Let us consider a translation cycle c_i composed of $\|c_i\|$ translation links. On an attribute basis, c_i 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_s and by ϵ_f the probability of p_1's translation (i.e., $T_{1\rightarrow 2}$) and another foreign translation (i.e., $T_{3\rightarrow 4} \ldots T_{n-1\rightarrow n}$) being wrong for the attribute in question. Considering the foreign error probabilities as being independent and identically distributed random variables, the probability of not having a foreign translation error along the cycle is

$$(1 - \epsilon_f)^{\|c_i\|-1}$$

Moreover, *compensating errors*, i.e., series of independent translation errors resulting in a correct mapping, may occur along the cycle of foreign links without being noticed by p_1 (which only has the final result q_n as its disposal). Thus, assuming $T_{1\rightarrow 2}$ correct and denoting by δ the (possibly high) probability of errors being compensated somehow, the probability of a cycle being positive is

$$(1-\epsilon_f)^{\|c_i\|-1} + (1-(1-\epsilon_f)^{\|c_i\|-1})\delta = p(\|c_i\|, \epsilon_f, \delta)$$

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

$$(1-(1-\epsilon_f)^{\|c_i\|-1})(1-\delta) = 1 - p(\|c_i\|, \epsilon_f, \delta).$$

Similarly, if we assume $T_{1\to 2}$ to be incorrect, the probability of a cycle being respectively negative and positive are

$$(1-\epsilon_f)^{\|c_i\|-1} + (1-(1-\epsilon_f)^{\|c_i\|-1})(1-\delta) = q(\|c_i\|, \epsilon_f, \delta)$$

and

$$(1-(1-\epsilon_f)^{\|c_i\|-1})\delta = (1 - q(\|c_i\|, \epsilon_f, \delta)).$$

Combining those equations, the likelihood of receiving a set of cycles $C = c_1, \ldots, c_k$, some positive ($c_i \in C^+$), other negative ($c_i \in C^-$), is

$$\begin{aligned} & l(c_1, \ldots, c_k) = \\ & \quad (1-\epsilon_s) \prod_{c_i \in C^+} p(\|c_i\|, \epsilon_f, \delta) \prod_{c_i \in C^-} (1 - p(\|c_i\|, \epsilon_f, \delta)) \\ & + \quad \epsilon_s \prod_{c_i \in C^-} q(\|c_i\|, \epsilon_f, \delta) \prod_{c_i \in C^+} (1 - q(\|c_i\|, \epsilon_f, \delta)) \end{aligned}$$

Now, we integrate over ϵ_f and δ, [2] and let ϵ_s tend toward zero and one in order to get the likelihood of the translation $T_{1\to 2}$ being semantically correct or incorrect respectively:

$$p_1 = \lim_{\epsilon_s \to 0} \int_{\delta=0}^{1} \int_{\epsilon_f=0}^{1} l(c_1 \ldots c_k)\, \mathrm{d}\epsilon_f\, \mathrm{d}\delta$$

$$p_2 = \lim_{\epsilon_s \to 1} \int_{\delta=0}^{1} \int_{\epsilon_f=0}^{1} l(c_1 \ldots c_k)\, \mathrm{d}\epsilon_f\, \mathrm{d}\delta$$

Finally, we define a variable γ for the relative degree of correctness of the translation:

$$\gamma = \frac{p_1}{p_1 + p_2}$$

Such an analysis may be performed for every outgoing link and every attribute independently, resulting in series of values γ_i^j indicating the likelihood of the translation T_j being correct for the attribute A_i. Examples of such calculations are given in Section 7.

As for the preceding section, we define now a feature vector and a similarity value to capture the semantic losses along the translation links. Let us suppose that a peer p_k issues a query $q = \pi_{ap}(\sigma_{p(as)}(\mu_{fa}(DB)))$ through a translation link $T_{k\to j}$. p_k computes a feature vector for q based on the cycle messages it has received as follows:

$$\overrightarrow{FV_{\circlearrowright}}(T_{k\to j}(q)) = (fv_{A1}^{\circlearrowright}, \ldots, fv_{Ak}^{\circlearrowright})$$

where the feature values $fv_{A_i}^{\circlearrowright}$ are defined for every attribute $A_i \in as \cup ap$ as

$$fv_{A_i}^{\circlearrowright}(T_{k\to j}(q)) = \gamma_i^j$$

These values are updated by iteratively multiplying the probabilities for each semantic domain traversed.[3] Thus, when forwarding a transformed query using a link $T_{k\to j}$, peer p_k updates each value $fv_{A_i}^{\circlearrowright}$ of the feature vector $\overrightarrow{FV_{\circlearrowright}}$ it has received along with the transformed query $T_{1\to\ldots\to k}(q)$ in this way:

$$\begin{aligned} & fv_{A_i}^{\circlearrowright}(T_{1\to\ldots\to k,k\to j}(q)) = \\ & \qquad fv_{A_i}^{\circlearrowright}(T_{1\to\ldots\to k}(q))\ fv_{A_i}^{\circlearrowright}(T_{k\to j}(T_{1\to\ldots\to k}(q))) \end{aligned}$$

where γ_i^j values for which we did not receive significant feedback (either because p does not have a representation for A_i or because no cycle message has been received so far) are evaluated to 1. The semantic similarity associated with this vector is

$$S_{\circlearrowright}(q, T_{1\to,\ldots,\to n}(q)) = \frac{\overrightarrow{W} \cdot \overrightarrow{FV_{\circlearrowright}}}{|\overrightarrow{W}|\ |\overrightarrow{FV_{\circlearrowright}}|}$$

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.

5.2 Result Analysis

The second mechanism for analyzing the semantic quality of a translation is based on the analysis of the results returned. Here, we assume that if peers agree on the semantics of their schema, they observe the same data dependencies. Therefore, for each functional dependency known, the peers analyze whether the values of the dependent attributes match. The percentage of matches then gives the degree of fidelity that can be put into the attributes involved in the functional dependency. To that end, the peers compare data objects received through a translation link to data objects they store themselves or that they receive through non-translation links (i.e., received from peers $\in N_e(p)$).

More precisely, consider two attributes A_i and A_j of p's schema S_p. Suppose now that A_i and A_j are tied to each other by a functional dependency $A_i \to A_j$; for every value $val(A_i)$ of A_i received in query responses from neighbor peers $\in N_d(p)$, p tries to find tuples sharing the same $val(A_i)$ in its own database or in returned tuples coming from peers $\in N_e(p)$. When common occurrences are found, p determines if tuples from peers $\in N_d(p)$ satisfy the functional dependency or not.

Thus the peer can maintain for every attribute A_i and for every outgoing translation link T_j a fidelity value κ_i^j keeping track of the proportion of tuples received from this link satisfying the functional dependency for the attributes A_i:[4]

$$\kappa_i^j = \frac{\#common\ tuples\ satisfying\ the\ FD}{\#common\ tuples}$$

[2] We could take into account density functions here if we have any *a priori* knowledge about those two variables.

[3] We consider here that if two translations Ta and Tb have probabilities of a and b respectively and are independent, the overall probability for $(Tb \circ Ta)$ to be correct is ab.

[4] κ_i is called *confidence measure* in the data mining terminology.

Assuming that the queries are properly processed and that they are properly distributed, the attribute fidelity eventually converges to the degree of compatibility among the databases (i.e., the percentage of data objects which match on dependent attributes for matching keys of a functional dependency).

In a way similar to what we defined for the cycles, we define a new feature vector with these values

$$\overrightarrow{FV_{\rightleftarrows}}(T_j(q)) = (fv_{A1}^{\rightleftarrows}, \ldots, fv_{Ak}^{\rightleftarrows})$$

whose feature values $fv_{A_i}^{\rightleftarrows}$ are defined for every attribute $A_i \in as \cup ap$ as

$$fv_{A_i}^{\rightleftarrows}(T_j(q)) = \kappa_i^j$$

and where, again, we evaluate missing values to 1 and we update the vectors iteratively:

$$fv_{A_i}^{\rightleftarrows}(T_{1\to\ldots\to k,k\to j}(q)) =$$

$$fv_{A_i}^{\rightleftarrows}(T_{1\to\ldots\to k}(q))\ fv_{A_i}^{\rightleftarrows}(T_{k\to j}(T_{1\to\ldots\to k}(q))).$$

The associated semantic similarity is, as expected:

$$S_{\rightleftarrows}(q, T_{1\to,\ldots,\to n}(q)) = \frac{\overrightarrow{W} \cdot \overrightarrow{FV_{\rightleftarrows}}}{\left|\overrightarrow{W}\right| \left|\overrightarrow{FV_{\rightleftarrows}}\right|}.$$

This value starts from one (in the semantic domain the query originates from) and approaches 0 as the query traverses more and more semantically heterogeneous domains.

6. GOSSIPING ALGORITHM

At this point, we have four measures (S_σ, S_π, $S_\circlearrowleft$ and $S_{\rightleftarrows}$) 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 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 $\overrightarrow{S_{min}}$ 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:

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

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 feature vectors: $similar(\overrightarrow{FV_i})$ evaluates to 1 if

   $$\frac{\overrightarrow{W} \cdot \overrightarrow{FV_i}}{\left|\overrightarrow{W}\right| \left|\overrightarrow{FV_i}\right|} \ \geq\ S_{min,i}$$

   that is if the semantic similarity is greater or equal to the specified minimal value, and to 0 otherwise
8. forward the query using the link if all $similar()$ tests succeed (i.e., evaluate to 1).

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

7. CASE STUDY

Several experiments were conducted following the approach presented above. This section presents one of them as a case study detailing how the aforementioned heuristics may be deployed in a concrete setting.

Seven people from our laboratory were first asked to design a simple XML document containing some project meta-data. The outcome of this voluntary 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 random graph connecting the different domains together with series of translation links (the resulting topology is depicted in Figure 4).

Translations were formulated as XQuery expressions in such a way that they strictly adhere to the principles stipulated above (see Section 3). As an example, Figure 5 presents two different documents as well as a simple query translation using the translation expression *T12*. Providing the authors with the required documents, we asked them to write the translations for every link departing from their domain (thus, p_A was asked to provide us with the translation to p_B, p_C and p_D). Finally, using the IPSI-XQ XQuery libraries [6] and the Xerces [15] XML parser, we built a query translator capable of handling and forwarding the queries following the gossiping algorithm.

We focus now on a single node, p_A, and on a single-attribute query issued by p_A to obtain all the titles of the different projects, namely:

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

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 S_σ and $S_{\rightleftarrows}$ for the rest of this study (S_σ always evaluates to 1 because there is no selection attribute, and so does $S_{\rightleftarrows}$ because we have not defined any

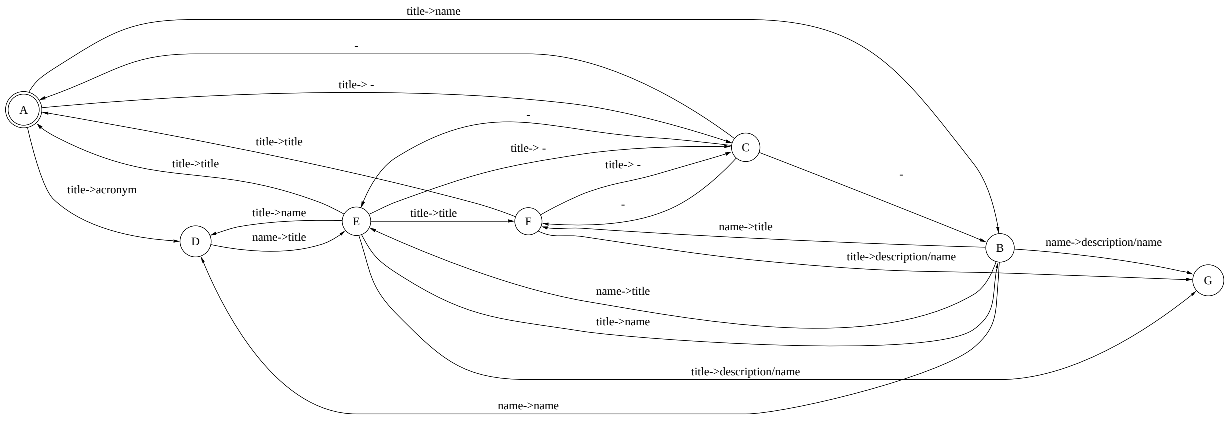
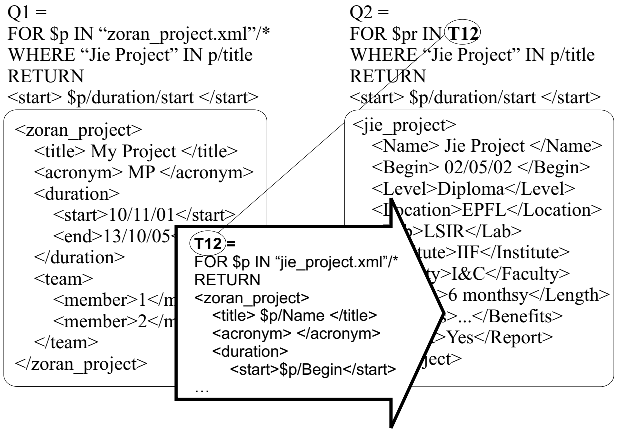


Figure 4: The Semantic Graph

functional dependency). The other minimal values are set to 0.5.



```
Q1 =
FOR $p IN "zoran_project.xml"/*
WHERE "Jie Project" IN p/title
RETURN
<start> $p/duration/start </start>
```

```
<zoran_project>
   <title> My Project </title>
   <acronym> MP </acronym>
   <duration>
      <start>10/11/01</start>
      <end>13/10/05
   </duration>
   <team>
      <member>1</m
      <member>2</m
   </team>
</zoran_project>
```

```
Q2 =
FOR $pr IN T12
WHERE "Jie Project" IN p/title
RETURN
<start> $p/duration/start </start>
```

```
<jie_project>
   <Name> Jie Project </Name>
   <Begin> 02/05/02 </Begin>
   <Level>Diploma</Level>
   <Location>EPFL</Location>
     >LSIR</Lab>
     tute>IIF</Institute>
     ty>I&C</Faculty>
     >6 monthsy</Length>
     >...</Benefits>
     >Yes</Report>
   ject>
```

```
T12 =
FOR $p IN "jie_project.xml"/*
RETURN
<zoran_project>
   <title> $p/Name </title>
   <acronym> </acronym>
   <duration>
      <start>$p/Begin</start>
...
```

Figure 5: The Translation Mechanism

All the domains have some representation for the title of the project (usually referred to as *name* or *title*, see Figure 4 where the translations for the attribute *title* are represented on top of the link), except p_C which only considers a mere *ID* for identifying the projects. Following the gossiping algorithm, p_A first attempts to transmit the query to its direct neighbors, i.e., p_B, p_C and p_D. p_B and p_D in turn forward the query to the other nodes, but p_C will in fact never receive the query: As p_C has no representation for the *title*, the only projection attribute would be lost in the translation process from p_A to p_C, lowering S_π to 0.

Let us now examine the semantic similarity $S_\circlearrowleft$. For the topology considered, thirty-one semantic cycles could be detected by p_A in the best case. As the query never traverses D, only eight cycles remain (Table 1 lists those cycles). We use now the formulas from Section 5; For its first outgoing link (i.e., the link going from p_A to p_B), p_A receives five positive cycles, raising the semantic similarity measure for this link and the attribute considered to 0.79.[5] p_A does not receive any semantically significant feedback for its second outgoing link $Tp_A \rightarrow p_B$, which is anyway handled by the syntactic analysis. Yet, it receives three negative cycles for its last outgoing link $Tp_A \rightarrow p_C$. This link is clearly semantically erroneous, mapping *title* onto *acronym*. This results in p_A excluding the link for this attribute, the semantic similarity dropping to 0.26.

Cycle	$T_{p_A \rightarrow p_C}$ Erroneous	$T_{p_B \rightarrow p_D}$ Erroneous
A, B, D, E, A	+	-
A, B, D, E, F, A	+	-
A, B, E, A	+	+
A, B, E, F, A	+	+
A, B, F, A	+	+
A, D, E, A	-	+
A, D, E, B, F, A	-	+
A, D, E, F, A	-	+

Table 1: Cycles Resulting In Positive(+) or Negative(-) Feedback

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

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

8. 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.

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

For example, in OBSERVER [10] 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, they use local measures for the loss of information when propagating queries and receiving results. Similarly to OBSERVER, KRAFT [14] proposes an agent-based architecture to manage ontological relationships in a distributed information system. Relationships among ontologies are expressed in a constraint language. [1] propose a model and architecture for managing distributed relational databases in a P2P environment. They 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 [13] a probabilistic framework for reasoning with assertions on schema relationships is introduced. Thus their approach deals with the problem of having possibly contradictory knowledge on schema relationships. [11] propose an architecture for the use of XML-based annotations in P2P systems to establish semantic interoperability.

EDUTELLA [12] 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.

Approaches for automatic schema matching—see [16] 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 [4] 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 [9], in which local relationships of information sources are exploited to derive global assessments on their quality (and eventually their meaning).

9. 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 fulfilment 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 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. 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.

10. REFERENCES

[1] P. 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.

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

[3] 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/.

[4] A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Learning to map between Ontologies on the Semantic Web. In *WWW2002*, 2002.

[5] D. Fensel, I. Horrocks, F. van Harmelen, S. Decker, Michael, Erdmann, and M. C. A. Klein. OIL in a Nutshell. In *EKAW*, pages 1–16, 2000.

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

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

[8] R. Hull. Managing Semantic Heterogeneity in Databases: A Theoretical Perspective. In *PODS*, pages 51–61, 1997.

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

[10] 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.

[11] A. Mukherjee, B. Esfandiari, and N. Arthorne. A Peer-to-peer System for Description and Discovery of Resource-sharing Communities. In *RESH*, 2002.

[12] W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmér, and T. Risch. EDUTELLA: a P2P networking infrastructure based on RDF. In *WWW2000*, pages 604–615, 2000.

[13] 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.

[14] A. D. Preece, K.-Y. Hui, W. Gray, T. J. M. B.-C. Philippe Marti, Z. Cui, and D. Jones. Kraft: An Agent Architecture for Knowledge Fusion. *IJCIS*, 10(1–2):171–195, 2001.

[15] A. X. Project. Xerces Parser, 2002.

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

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