Small Can Be Beautiful in the Semantic Web

Marie-Christine Rousset

LRI, CNRS – Université Paris Sud XI – INRIA Futurs,
Bâtiment 490, Université Paris Sud XI, 91405 Orsay cedex, France

Abstract. In 1984, Peter Patel-Schneider published a paper [1] entitled *Small can be Beautiful in Knowledge Representation* in which he advocated for limiting the expressive power of knowledge representation formalisms in order to guarantee good computational properties and thus make knowledge-based systems usable as part of larger systems. In this paper, I aim at showing that the same argument holds for the Semantic Web: if we want to give a chance for the Semantic Web to scale up and to be broadly used, we have to limit the expressive power of the ontologies serving as semantic marking up of resources. In addition, due to the scale of the Web and the disparity of its users, it is unavoidable to have to deal with distributed heterogeneous ontologies. In this paper, I will argue that a peer-to-peer infrastructure enriched with simple distributed ontologies (e.g., taxonomies) reconciled through mappings is appropriate and scalable for supporting the future Semantic Web.

1 Introduction

The Semantic Web [2] envisions a world-wide distributed architecture where data and computational resources will easily inter-operate based on semantic marking up of web resources using *ontologies*. Ontologies are a formalization of the semantics of application domains (e.g., tourism, biology, medecine) through the definition of classes and relations modeling the domain objects and properties that are considered as meaningful for the application. Building ontologies is a hard and time-consuming task for which there exists only some very general principles to guide the ontology designers [3,4,5] who still have to face with many modeling choices. Even for a same domain, different modeling choices can lead to very different ontologies. In particular, the choice of the basic classes and relations for modeling the basic domain vocabulary is subject to many variations depending on the ontology designers. The appropriate level of detail of the ontology descriptions is not easy to determine either and mainly depends on the purpose of the ontology construction.

Several important issues remain open concerning the building and usage of ontologies in the setting of the Semantic Web. In this talk, I will discuss some of them and I will explain my vision of the kind of infrastructure that I consider as scalable for supporting the future Semantic Web. An instance of that infrastructure is implemented in the Somewhere [6] system that I will present briefly.

S.A. McIlraith et al. (Eds.): ISWC 2004, LNCS 3298, pp. 6–16, 2004.

2 Simple Versus Complex Ontologies

A first question concerns the choice of the right expressive power for an adequate modeling of the ontologies needed in the Semantic Web. The current tendancy promoted by the knowledge representation community is to consider that formal languages with high expressive power are required because they enable a fine-grained description of domain ontologies. Such a choice is questionable for several reasons. The main argument that I would oppose against this choice is algorithmic: well-known complexity results show that exploiting ontologies modeled using expressive formal languages (e.g., OWL [7]) cannot scale up to complex or large ontologies. Another argument is that formal languages with high expressive power are difficult to handle for users having to model an application domain. In the Semantic Web setting, the purpose of ontologies is to serve as a semantic markup for web resources and as a support for querying in order to obtain more efficient and precise search engines. Therefore, they must be simple to be correctly understood and rightly exploited by humans (users or experts), and they must be expressible in a formal language for which the algorithmic complexity of reasoning is reasonable to make them really machine processable. Taxonomies of atomic classes are examples of simple ontologies that I envision as good candidates for serving as marking up resources at the Web scale. They are easy to understand for users and practitioners. They may even be automatically constructed by data or text mining. They are conform to the W3C recommandation being a subset of OWL-DL that we could call OWL-PL since they can be encoded in propositional logic. As a result, query answering becomes feasible at a large scale, which is the goal to reach eventually if we want the Semantic Web to become a reality.

3 Personalized and Distributed Versus Standardized Ontologies

Another question deals with the possibility of building consensual domain ontologies that would be broadly shared by users over the Web for marking up their data or resources. Building a universal ontology (or using an existing one, e.g., CYC [8]) that could serve as a reference set of semantic tags for labelling the data and documents of the Web is, at worst an utopia, at best an enormous enterprise which may eventually turn out to be useless in practice for the Semantic Web. The current tendency (e.g., [9,10,11]) consists in building standardized ontologies per domain. So far, it is an open question whether such an approach is likely to scale up to the Web because it cannot be taken as granted that users will appropriate those ontologies. The risk is that ontology designers spend a lot of time building an expected consensual ontology which will not be eventually broadly used because end users will prefer to use their own ontologies. In the same way as we have our own view of the nesting and the names of the different folders for structuring our personal file systems, mail files or bookmarks, it is likely that people will prefer using their own ontology to mark up the data or resources they agree to make available through the Semantic Web. There is a little chance that they will accept to use an external ontology which they are not used to for re-labelling the resources that they have already marked-up.

However, if the ontologies are simple (e.g., taxonomies of terms) users may be ready to establish mappings between their own ontology and ontologies of some users with whom they share some topics of interest.

Consider for instance a user Bob found of music who has already stored plenty of music files in three folders that he has named $Classic$, $Jazz$ and $Rock$ to distinguish pieces of classical music, jazz and rock respectively. Suppose now that by searching new music files on the web, he discovers a user Tom making available on his web page music files but marked up by the name of the composers. Establishing a mapping saying that what Tom marks as $Mozart$ (pieces of music) is a specialization of his own view of $Classic$ (pieces of music) is straighforward for Bob. This mapping will make possible for any user querying the web using Bob's ontology to get $Mozart$ files stored at the Tom's web page as an answer to his search for classical pieces of music.

In this vision of the Semantic Web introduced in [12], no user imposes to others his own ontology but logical mappings between ontologies make possible the creation of a web of people in which personalized semantic marking up of data cohabits nicely with a collaborative exchange of data. In this view, the Web is a huge Peer Data Management System based on simple distributed ontologies and mappings.

4 Peer Data Management Systems

Peer data management systems (PDMS) have been proposed recently [13,14,15,16] to generalize the centralized approach of information integration systems based on single mediators, in which heterogeneous data sources are reconciled through logical mappings between each source schema or ontology and a single mediated schema or ontology. A centralized vision of mediation is appropriate for building semantic portals dedicated to a given domain but is too rigid to scale up to the Semantic Web for which PDMSs based on distributed mediation are more adapted. In a PDMS, there is no central mediator: each peer has its own ontology and data or services, and can mediate with some other peers to ask and answer queries. The existing PDMSs vary according to (a) the expressive power of their underlying data model and (b) the way the different peers are semantically connected. Both characteristics have impact on the allowed queries and their distributed processing.

In Edutella [17], each peer stores locally data (educational resources) that are described in RDF relatively to some reference ontologies (e.g., dmoz [9]). For instance, a peer can declare that it has data corresponding to the concept of the dmoz taxonomy corresponding to the path *Computers/Programming/Languages/Java*, and that for such data it can export the *author* and the *date* properties. The overlay network underlying Edutella is a hypercub of super-peers to which peers are directly connected. Each super-peer is a mediator over the data of the peers connected to it. When it is asked, its first task is to check if the query matches with its schema: if that is the case, it transmits the query to the peers connected to it, which are likely to store the data answering the query ; otherwise, it routes the query to some of its neighbour super-peers according to a strategy exploiting the hypercub topology for guaranteeing a worst-case logarithmic time for reaching the relevant super-peer.

In contrast with Edutella, Piazza [13,18] does not consider that the data distributed over the different peers must be described relatively to some existing reference schemas. In Piazza, each peer has its own data and schema and can mediate with some other peers by declaring *mappings* between its schema and the schemas of those peers. The topology of the network is not fixed (as the super-peers hypercub in Edutella) but accounts for the existence of mappings between peers: two peers are logically connected if there exists a mapping between their two schemas. The underlying data model considered in the first version of Piazza [13] is relational and the mappings between relational peer schemas are inclusion or equivalence statements between conjunctive queries. Such a mapping formalism encompasses the *local-as-views* and the *global-as-views* formalisms used in information integration based on centralized single mediators for stating the mappings between the schemas of the data sources to be integrated and the mediated schema. The price to pay is that query answering is undecidable except if some restrictions are imposed on the mappings or on the topology of the network ([13]). The currently implemented version of Piazza [18] relies on a tree-based data model: the data is in XML and the mappings are equivalence and inclusion statements between XML queries. Query answering is implemented based on practical (not complete) algorithms for XML query containment and rewritings. The scalability of Piazza so far does not go up to more than about eighty peers in the published experiments and relies on a wide range of optimizations (mappings composition [19], paths pruning [20]), made possible by the centralized storage of all the schemas and mappings in a global server.

In Somewhere [6], we have made the choice of being fully distributed: there are neither super-peers (as in Edutella) nor a central server having the global view of the overlay network as in Piazza. In addition, we aim at scaling up to thousands of peers. For making it possible, we have chosen a simple class-based data model in which the data is a set of resource identifiers (URIs or URLs), the schemas are (simple) definitions of classes possibly constrained by inclusion, disjunction or equivalence statements, and mappings are inclusion, disjunction or equivalence statements between classes of different peer schemas. That data model is in accordance with the W3C recommandations since it is captured by the propositional fragment of the OWL ontology language.

5 Overview of the Somewhere PDMS

This section reports a joint work [6] with Philippe Adjiman, Philippe Chatalic, Francois Goasdoué and Laurent Simon.

5.1 Data Model

In Somewhere a new peer joins the network through some peers that it knows (its acquaintances) by declaring mappings between its own ontology and the ontologies of its acquaintances. Queries are posed to a given peer using its local ontology. The answers that are expected are not only instances of local classes but possibly instances of classes of peers distant from the queried peer if it can be infered from the peer ontologies and the mappings that they satisfy the query. Local ontologies, storage descriptions and mappings are defined using a fragment of OWL DL which is the description logics fragment of the Ontology Web Language recommended by W3C. We call OWL-PL the fragment of OWL-DL that we consider in Somewhere, where PL stands for propositional logic. OWL PL is the fragment of OWL DL reduced to the union ($\sqcup$), intersection ($\sqcap$) and complement ($\neg$) constructors for building class descriptions.

Peer ontologies: Each peer ontology is made of a set of axioms of class (partial or complete) definitions that associate class identifiers with class descriptions, and disjointness, equivalence or inclusion statements between class descriptions. Class identifiers are unique to each peer: we use the notation $P{:}CI$ for a class identifier CI of the ontology of a peer P.

Peer storage descriptions: The specification of the data that is stored locally in a peer P is done through assertional statements relating data identifiers (e.g., URLs or URIs) to class identifiers of the ontology of the peer P. The class identifiers of P involved in such statements are called the *extensional classes* of P and their extensions are the sets of data identifiers associated with them.

Mappings: Mappings are disjointness, equivalence or inclusion statements involving classes of different peers. They express the semantic correspondence that may exist between the ontologies of different peers.

Schema of a Somewhere peer-to-peer network: In a Somewhere network, the schema is not centralized but distributed through the union of the different peer ontologies and the mappings. The important point is that each peer has a partial knowledge of the schema: it just knows its own data and local ontology, and mappings with its acquaintances.

Let $\mathcal{P}$ be a Somewhere peer-to-peer network made of a collection of peers $\{P_i\}_{i\in[1..n]}$. For each peer P_i, let O_i and M_i be respectively the local ontology of P_i and the set of mappings stated at P_i between classes of O_i and classes of the ontologies of the acquaintances of P_i. The schema $\mathcal{S}$ of $\mathcal{P}$ is the union $\bigcup_{i\in[1..n]} O_i \cup M_i$ of the ontologies and of the sets of mappings of all the peers composing $\mathcal{P}$.

Semantics: The semantics is a standard logical formal semantics defined in terms of *interpretations*. An interpretation I is a pair $(\Delta^I, .^I)$ where Δ is a non-empty set, called the domain of interpretation, and $.^I$ is an interpretation function which assigns a subset of Δ^I to every class identifier and an element of Δ^I to every data identifier.

An interpretation I is a *model* of the distributed schema of a Somewhere peer-to-peer network $\mathcal{P} = \{P_i\}_{i\in[1..n]}$ iff each axiom in $\bigcup_{i\in[1..n]} O_i \cup M_i$ is satisfied by I.

Interpretations of axioms rely on interpretations of class descriptions which are inductively defined as follows:

- $(C_1 \sqcup C_2)^I = C_1^I \cup C_2^I$
- $(C_1 \sqcap C_2)^I = C_1^I \cap C_2^I$
- $(\neg C)^I = \Delta^I \backslash C^I$

Axioms are satisfied if the following holds:

- $C \sqsubseteq D$ is satisfied in I iff $C^I \subseteq D^I$
- $C \equiv D$ is satisfied in I iff $C^I = D^I$
- $C \sqcap D \equiv \bot$ is satisfied in I iff $C^I \cap D^I = \emptyset$
- $C(a)$ is satisfied in I iff $a^I \in C^I$

A Somewhere peer-to-peer network is *satisfiable* iff its (distributed) schema has a model.

Given a Somewhere peer-to-peer network $\mathcal{P} = \{P_i\}_{i\in[1..n]}$, a class description C *subsumes* a class description D iff in each model I of the schema of $\mathcal{P}$, $D^I \subseteq C^I$.

5.2 Illustrative Example of the Somewhere Data Model

Let us consider four persons Ann, Bob, Chris and Dora, each of them bookmarking URLs about restaurants they know or like. Each has his/her own taxonomy for categorizing restaurants. In addition, they have to describe their stored available data, i.e., the sets of URLs that they accept to make available to the PDMS. They do it by declaring some *extensional classes* (denoted $P{:}ViewX$) as subclasses of some classes ($P{:}X$) of their ontology and by assigning to them the set of corresponding URLs as their instances.

Ann, who is working as a restaurant critics, organizes its restaurant URLs according to the following classes:

- the class $Ann{:}G$ of restaurants considered as offering a "good" cooking, among which she distinguishes the subclass $Ann{:}R$ of those which are rated:
  $Ann{:}R \sqsubseteq Ann{:}G$.
- the class $Ann{:}R$ is the union of three disjoint classes $Ann{:}S1$, $Ann{:}S2$, $Ann{:}S3$ corresponding respectively to the restaurants rated with $1, 2$ or 3 stars:
  $Ann{:}R \equiv Ann{:}S1 \sqcup Ann{:}S2 \sqcup Ann{:}S3$
  $Ann{:}S1 \sqcap Ann{:}S2 \equiv \bot \quad Ann{:}S1 \sqcap Ann{:}S3 \equiv \bot \quad Ann{:}S2 \sqcap Ann{:}S3 \equiv \bot$
- the classes $Ann{:}I$ and $Ann{:}O$, respectively corresponding to Indian and Oriental restaurants,
- the classes $Ann{:}C$, $Ann{:}T$ and $Ann{:}V$ which are subclasses of $Ann{:}O$ denoting Chinese, Taï and Vietnamese restaurants respectively:
  $Ann{:}C \sqsubseteq Ann{:}O$, $Ann{:}T \sqsubseteq Ann{:}O$, $Ann{:}V \sqsubseteq Ann{:}O$

Suppose that the data stored by Ann she accepts to make available are data on restaurants of various specialties, but that among the rated restaurants she stores and makes available those rated with 2 stars. The extensional classes declared by Ann are then:

$Ann{:}ViewS2 \sqsubseteq Ann{:}S2$, $Ann{:}ViewC \sqsubseteq Ann{:}C$
$Ann{:}ViewV \sqsubseteq Ann{:}V$, $Ann{:}ViewT \sqsubseteq Ann{:}T$
$Ann{:}ViewI \sqsubseteq Ann{:}I$

Bob, who is found of Asian cooking and likes high quality, organizes his restaurant URLs according to the following classes:

- the class $Bob{:}A$ of Asian restaurants,
- the class $Bob{:}Q$ of restaurants of high quality that he knows.

Suppose that he wants to make available every data that he has stored. The extensional classes that he declares are $Bob{:}ViewA$ and $Bob{:}ViewQ$ (as subclasses of $Bob{:}A$ and $Bob{:}Q$):

$Bob{:}ViewA \sqsubseteq Bob{:}A$, $Bob{:}ViewQ \sqsubseteq Bob{:}Q$

Chris is more found of fish restaurants but recently discovered some places serving a very nice cantonese cuisine. He organizes its data with respect to the following classes:

- the class $Chris{:}F$ of fish restaurants,
- the class $Chris{:}CA$ of Cantonese restaurants

Suppose that he declares the extensional classes $Chris{:}ViewF$ and $Chris{:}ViewCA$ as subclasses of $Chris{:}F$ and $Chris{:}CA$ respectively:

$Chris{:}ViewF \sqsubseteq Chris{:}F$, $Chris{:}ViewCA \sqsubseteq Chris{:}CA$

Finally, Dora organizes her restaurants URLs around the class $Dora{:}DP$ of her preferred restaurants, among which she distinguishes the subclass $Dora{:}P$ of pizzerias and the subclass $Dora{:}SF$ of seafood restaurants.

Suppose that the only URLs that she stores concerns pizzerias: the only extensional class that she has to declare is $Dora{:}ViewP$, as a subclass of $Dora{:}P$:

$Dora{:}ViewP \sqsubseteq Dora{:}P$

Ann, Bob, Chris and Dora are modelled as four peers. Each of them is able to express what he/she knows about others using mappings stating properties of class inclusion or equivalence.

Ann is very confident in Bob's taste and agrees to include Bob'selection as good restaurants by stating $Bob{:}Q \sqsubseteq Ann{:}G$. Finally, she thinks that Bob's Asian restaurants encompass her concept of Oriental restaurant : $Ann{:}O \sqsubseteq Bob{:}A$.

Bob: Bob knows that what he calls Asian cooking corresponds exactly to what Ann classifies as Oriental cooking. This may be expressed using the equivalence statement : $Bob{:}A \equiv Ann{:}O$ (note the difference of perception of Bob and Ann regarding the mappings between $Bob{:}A$ and $Ann{:}O$).

Chris: Chris considers that what he calls fish specialties is a particular case of what Dora calls seafood specialties : $Chris{:}F \sqsubseteq Dora{:}SF$

Dora: Dora counts on both Ann and Bob to obtain good Asian restaurants :
$Bob{:}A \sqcap Ann{:}G \sqsubseteq Dora{:}DP$

Figure 1 describes the peer network induced by the mappings. In order to alleviate the notations, we omit the local peer name prefix except for the mappings. Edges are labeled with the class identifiers that are shared through the mappings between peers.

5.3 Query Answering

Queries and answers: Queries are combinations of classes of a given peer ontology. The corresponding answer sets are expressed in intention in terms of the combinations of extensional classes that are *rewritings* of the query. The point is that extensional classes

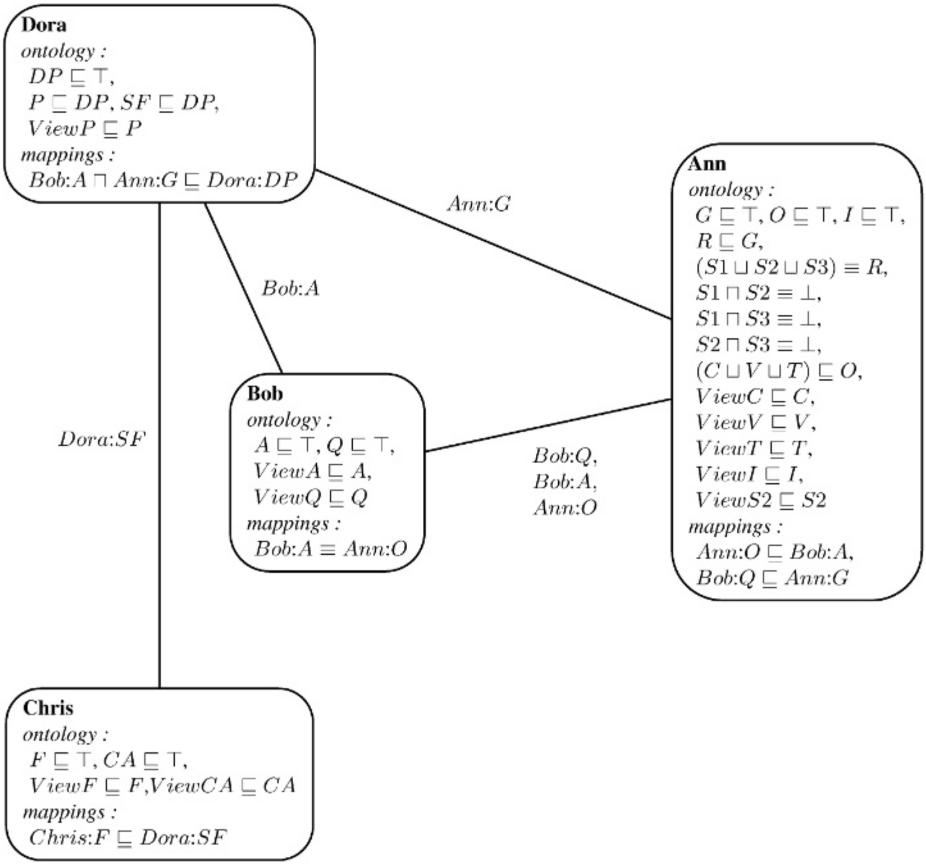

Fig. 1. The restaurants PDMS

of distant peers can participate to the rewritings and thus their instances to the answer set of a query posed to a given peer.

Given a Somewhere peer-to-peer network $\mathcal{P} = \{P_i\}_{i \in [1..n]}$, a logical combination Q_e of extensional classes is a *rewriting* of a query Q iff Q subsumes Q_e. Q_e is a *maximal rewriting* if there does not exist another rewriting Q'_e of Q subsuming it.

In the Somewhere setting, query rewriting can be equivalently reduced to distributed reasoning over logical propositional theories by a straighforward propositional encoding of the distributed ontologies and mappings composing the distributed schema of a Somewhere peer-to-peer network.

Propositional encoding: The propositional encoding concerns the schema of a Somewhere peer-to-peer network and the queries. It consists in transforming each query and schema statement into a propositional formula using class identifiers as propositional variables. The propositional encoding of a class description D, and thus of a query, is the propositional formula $Prop(D)$ obtained inductively as follows:

- $Prop(A) = A$, if A is a class identifier
- $Prop(D_1 \sqcap D_2) = Prop(D_1) \wedge Prop(D_2)$
- $Prop(D_1 \sqcup D_2) = Prop(D_1) \vee Prop(D_2)$
- $Prop(\neg D) = \neg(Prop(D))$

The propositional encoding of the schema $\mathcal{S}$ of a Somewhere peer-to-peer network $\mathcal{P}$ is the distributed propositional theory $Prop(\mathcal{S})$ made of the formulas obtained inductively from the axioms in $\mathcal{S}$ as follows:

- $Prop(C \sqsubseteq D)$ = $Prop(C) \Rightarrow Prop(D)$
- $Prop(C \equiv D)$ = $Prop(C) \Leftrightarrow Prop(D)$
- $Prop(C \sqcap D \equiv \bot)$= $\neg Prop(C) \vee \neg Prop(D)$

That propositional encoding transfers satisfiability and maps (maximal) conjunctive rewritings of a query Q to clausal proper (prime) implicates of the propositional formula $\neg Prop(Q)$.

Therefore, we can use the message passing algorithm presented in [21] for query rewriting in Somewhere. That algorithm is the first consequence finding algorithm in a peer-to-peer setting: it is anytime and computes consequences gradually from the solicited peer to peers that are more and more distant.

We illustrate the distributed resulting query processing on the example in Section 5.2. Consider that a user queries the restaurants PDMS through the **Dora** peer by asking the query $Dora{:}DP$, meaning that he is interested in getting as answers the set of favourite restaurants of Dora:

- He will get as a first rewriting $Dora{:}ViewP$ corresponding to the extensional class of the URLs of pizzerias stored locally by Dora.
- Then, the mapping $Chris{:}F \sqsubseteq Dora{:}SF$ leads to a new rewriting, $Chris{:}ViewF$, meaning that a way to get restaurants liked by Dora is to obtain the Fish restaurants stored by Chris.
- Finally, the mapping $Bob{:}A \sqcap Ann{:}G \sqsubseteq Dora{:}DP$ leads to the splitting of $Bob{:}A \sqcap Ann{:}G$ into the two subqueries $Bob{:}A$ and $Ann{:}G$; they are transmitted respectively to the peers Bob and Ann, which process them independently:
  - $Bob{:}ViewA$ is a local rewriting of $Bob{:}A$, which is transmitted back to the Dora peer, where it is queued for a future combination with rewritings of the other subquery $Ann{:}G$. In addition, guided by the mapping $Ann{:}O \equiv Bob{:}A$, the Bob peer transmits to the Ann peer the query $Ann{:}O$; the Ann peer processes that query locally and transmits back to the Bob peer the rewriting: $Ann{:}ViewC \sqcup Ann{:}ViewT \sqcup Ann{:}ViewV$, which in turn is transmitted back to the Dora peer as an additional rewriting for the subquery $Bob{:}A$ and queued there,
  - $Ann{:}ViewS2$ is a local rewriting of $Ann{:}G$, which is transmitted back to the Dora peer, and combined there with the two queued rewritings of $Bob{:}A$ ($Bob{:}ViewA$ and $Ann{:}ViewC \sqcup Ann{:}ViewT \sqcup Ann{:}ViewV$).

As a result, two rewrirings are sent back to the user:
- $Ann{:}ViewS2\sqcap\ Bob{:}ViewA$, meaning that a way to obtain restaurants liked by Dora is to find restautants that are both stored by Ann as rated with 2 stars and by Bob as Asian restaurants,
- $Ann{:}ViewS2\sqcap\ (ViewC\sqcup\ Ann{:}ViewT\sqcup\ Ann{:}ViewV)$, meaning that another way to obtain restaurants liked by Dora is to find restautants stored by Ann as restaurants rated with 2 stars and also as Chinese, Thai or Vietnamese restaurants. Note that this rewriting, which is obtained after several splitting and re-combining turns out to be composed of extensional classes of the same peer (Ann).

- Because of the mapping $Bob{:}Q \sqsubseteq Ann{:}G$, Ann transmits the query $Bob{:}Q$ to Bob, which transmits back to Ann $Bob{:}ViewQ$ as a rewriting of $Bob{:}Q$ and then of $Ann{:}G$. Ann then transmits $Bob{:}ViewQ$ back to Dora as a rewriting of $Ann{:}G$. At Dora's side, $Bob{:}ViewQ$ is now combined with the queued rewritings of $Bob{:}A$ ($Bob{:}ViewA$ and $Ann{:}ViewC\sqcup Ann{:}ViewT\sqcup Ann{:}ViewV$). As a result, two new rewrirings are sent back to the user:
- $Bob{:}ViewQ\sqcap\ Bob{:}ViewA$, meaning that to obtain restaurants liked by Dora you can take the restaurants that Bob stores as high quality restaurants and also as Asian restaurants,
- $Bob{:}ViewQ\sqcap\ (Ann{:}ViewC\sqcup\ Ann{:}ViewT\sqcup\ Ann{:}ViewV)$, providing a new way of getting restaurants liked by Dora (restaurants that are both considered as high quality restaurants by Bob and stored as Chinese, Thai or Vietnamese restaurants).

A peer-to-peer architecture implementing this distributed query rewriting algorithm has been developed and the first experimental results of its scalability are promising [6]. This architecture is used in a joint project with France Télécom, which aims at enriching peer-to-peer web applications with semantics in the form of taxonomies (e.g., Someone [12]).

6 Conclusion

Most of the concepts, tools and techniques deployed so far by the Semantic Web community correspond to the "big is beautiful" idea that high expressivity is needed for describing domain ontologies. As a result, when they are applied, the so-called Semantic Web technologies are mostly used for building thematic portals but do not scale up to the Web.

In this paper, I have argued in favour of a "simple-is-beautiful" vision of the Semantic Web consisting in progressing step by step from the current web towards a more semantic web. The first challenging step (which is far being reached) should be to do best than Google for searching through the whole Web. My vision of a "Semantic Google" would be to replace the use of words for annotating web documents by terms of a taxonomy. Though terms of a taxonomy are words, the (big) difference is that the taxonomy provides a kind of context of interpretation for those terms which is most of the time sufficient in practice to desambiguate their meaning. Therefore, it is important that taxonomies whose terms are used for annotating web resources are attached to those web resources. In this vision, any user is allowed to annotate freely web resources with terms of the taxonomies of his choice but he must attach those taxonomies to the web resources he has annotated. The glue of such a semantic web would be provided by mappings between taxonomies, and the infrastructure implementing it would be a peer-to-peer one.

References

1. Patel-Schneider, P.: Small can be beautiful in knowledge representation. In: IEEE Workshop on Principles of Knowledge-Based Systems. (1984)
2. Berners-Lee, T., Hendler, J., O.Lassila: The semantic web. Scientific American, 279 (2001)
3. Sure, Y., Staab, S., Studer, R.: Methodology for development and employment of ontology based knowledge management applications. SIGMOD Record **34** (2002) 8–24
4. Studer, R., Benjamins, R., Fensel, D.: Knowledge engineering: Principles and methods. Data and Knowledge Engineering **25** (1998) 161–197
5. Fernandez-Lopez, M., Gomez-Perez, A.: Overview and analysis of methodologies for building ontologies. Knowledge Engineering Review **17** (2002) 129–156
6. Adjiman, P., Chatalic, P., Goasdoué, F., Rousset, M.C., Simon, L.: Somewhere in the semantic web. Technical report, http://www.lri.fr/∼goasdoue/biblio/ACGRS-TR-SW.pdf (2004)
7. W3C: Owl web ontology language semantics and abstract syntax (2004)
8. Lenat, D.: Cyc: A large-scale investment in knowledge infrastructure. Communication of the ACM **38** (1995) 32–38
9. Dmoz: The open directory project (http://dmoz.org/)
10. OTA: Open travel alliance (http://www.opentravel.org/)
11. IEEE: Ieee learning object metadata (http://ltsc.ieee.org/wg12)
12. Plu, M., Bellec, P., Agosto, L., van de Velde, W.: The web of people: A dual view on the WWW. In: Int. World Wide Web Conf. (2003)
13. Halevy, A., Ives, Z., Suciu, D., Tatarinov, I.: Schema mediation in peer data management systems. In: ICDE'03. (2003)
14. Ooi, B., Shu, Y., Tan, K.L.: Relational data sharing in peer data management systems. **23** (2003)
15. Arenas, M., Kantere, V., Kementsietsidis, A., Kiringa, I., Miller, R., Mylopoulos, J.: The hyperion project: From data integration to data coordination. (2003)
16. Bernstein, P., Giunchiglia, F., Kementsietsidis, A., Mylopoulos, J., Serafini, L., Zaihraheu, I.: Data management for peer-to-peer computing: A vision. In: WebDB Workshop. (2002)
17. Nedjl, W., Wolf, B., Qu, C., Decker, S., Sintek, M., Naeve, A., Nilsson, M., Palmer, M., Risch, T.: Edutella: a p2p networking infrastructure based on rdf. In: WWW'02. (2002)
18. Halevy, A., Ives, Z., Tatarinov, I., Mork, P.: Piazza: data management infrastructure for semantic web applications. In: WWW'03. (2003)
19. Madhavan, J., Halevy, A.: Composing mappings among data sources. In: VLDB 03. (2003)
20. Tatarinov, I., Halevy, A.: Efficient query reformulation in peer data management systems. In: SIGMOD 04. (2004)
21. Adjiman, P., Chatalic, P., Goasdoué, F., Rousset, M.C., Simon, L.: Distributed reasoning in a peer-to-peer setting. In: ECAI 04. (2004)