# Query Reformulation over Ontology-based Peers (Extended Abstract)*

*Diego Calvanese*[1], *Giuseppe De Giacomo*[2], *Domenico Lembo*[2], *Maurizio Lenzerini*[2], and *Riccardo Rosati*[2]

[1] Faculty of Computer Science, Free University of Bolzano/Bozen,
calvanese@inf.unibz.it
[2] Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza",
*lastname*@dis.uniroma1.it

## 1 Introduction

Recently, the issue of integration and cooperation between information nodes in a networked environment has been studied in different contexts, as data integration [9], the Semantic Web [7], Peer-to-Peer [1, 6], Grid, and service oriented computing [13, 8].

Put in an abstract way, these systems are characterized by an architecture constituted by various autonomous nodes (called sites, sources, agents, or, as we do here, peers) which hold information, and which are linked to other nodes by means of mappings. Two basic problems arising in this architecture are: how to discover and express the mappings between peers [6, 1], and how to exploit the mappings in order to answer queries posed to one peer [9, 12]. The latter is the problem studied in this paper.

Although several interesting results have been reported in each of the above contexts, we argue that a deep understanding of the problem of answering queries in a networked environment is still lacking, in particular when the information in each peer is modeled in terms of a knowledge base.

The goal of this paper is to present some basic, fundamental results on this problem. We consider a simplified setting based on just two peers, called local and remote, respectively. Suitable mappings relate information in the remote peer to the information in the local peer. We assume that the queries to be answered are posed to the local peer, and that each of the two peers provides the service of answering queries expressed over its underlying knowledge base, with these two services being the only basic services that we can rely upon. Thus, the problem we address in the paper, called the "What-To-Ask" problem (WTA), is to find a way to answer queries posed to the local peer by relying only on the two query answering services available at the peers. We study this problem in a first-order logic (FOL) context and present the following contributions.

(1) In Section 2, we formalize the above mentioned architecture, we provide its semantics, the semantics of query answering, and the formal definition of WTA.

(2) In Section 3, we specialize the general framework to the case where a basic ontology language is used to express the knowledge bases of the two peers. We show that in this case there is an algorithm that allows to solve any instance of WTA, i.e., that allows to compute what we should ask to the remote peer in order to answer a query posed to the local peer.

(3) In Section 4, we show that, if we slightly enrich the expressive power of the ontology language, WTA does not always admit a solution.

* This paper is an extended abstract of [3]

The problem studied here is crucial in several contexts. In particular, it is important in the Semantic Web, where query reformulation over ontologies has been investigated (see, e.g., [4, 16]). It is also related to query answering in P2P and Grid computing. However, in such architectures, peers are in general assumed to be databases, or query languages much less expressive than those considered here are adopted. A notable exception is [14], where knowledge-based peers are studied, but the problem addressed there is different from our reformulation problem. Finally, the algorithm presented in Section 3 can be seen as a new query rewriting algorithm for data integration systems [9], or as a procedure for the "composition" of the query answering services provided by the peers of a service-oriented architecture (in the line of [15], where, however, peers do not have constraints). As far as we know, our work is the first to deal with query reformulation by relying only on the query answering services of the peers.

## 2 The Framework

We now set up a formal framework for *knowledge-based peer* interoperation. We first introduce some preliminary notions. We assume that the domain of interpretation is a fixed denumerable set of elements $\Delta$ and that every such element is denoted uniquely by a constant $c$, called its *standard name* [10]. We denote by $\Gamma$ the standard names.

Formally, a peer is a tuple of the form $P = \langle K, V, M\rangle$ where: $K$ is a knowledge base written in some subset of FOL on the alphabet formed by the standard names as constants, and a set of relation names (we do not consider functions in this paper); $V$ is the exported view of $K$ (i.e., a knowledge base formed as a subset of $K$); and $M$ is a set of mapping assertions whose form will be shown below.

Clients can *ask* queries to the peer $P$, as long as such queries can be accepted by $P$. A query is *accepted by* $P$ if it is a query expressed in the subset of FOL (possibly with equalities) over the alphabet of $V$, which is supported by $P$. We remind that a *FOL query* is an open formula of the form $\{x_1, \ldots, x_n \mid \phi(x_1, \ldots, x_n)\}$ where $x_1, \ldots, x_n$ are the free variables of $\phi$, and $n$ is the *arity* of the query.

A knowledge-based peer system is formed by several peers sharing domain of interpretation and standard names. Here we concentrate on knowledge-based systems of a very specific form. They consist of only two peers $P_\ell = \langle K_\ell, V_\ell, M_\ell\rangle$, called *local peer*, which is the peer to which the client is connected, and $P_r = \langle K_r, V_r, \emptyset\rangle$, called the *remote peer*. Observe that the remote peer does not contain mapping assertions.

The mapping $M_\ell$ in the local peer is constituted by a finite set of *assertions* of the form $q_r \rightsquigarrow q_\ell$, where $q_r$ and $q_\ell$ are two queries of the same arity, called remote and local query, respectively. The local query $q_\ell$ is expressed in some FOL query language over $K_\ell$, while the remote query $q_r$ must be a query accepted by $P_r$.

A mapping assertion $q_r \rightsquigarrow q_\ell$ has an immediate interpretation in FOL: it states that $\forall x_1, \ldots, x_n.\phi_r(x_1, \ldots, x_n) \supset \phi_\ell(x_1, \ldots, x_n)$, where $\phi_\ell$ and $\phi_r$ are the open formulas constituting the queries $q_\ell$ and $q_r$, respectively.

Given a FOL query $q$ of arity $n$ over a FOL theory $\mathcal{T}$, we indicate with $cert(q, \mathcal{T})$ the *certain answers* to $q$ over $\mathcal{T}$, i.e., the set of tuples of constants of $\Gamma$ such that: $cert(q, \mathcal{T}) = \{(c_1, \ldots, c_n) \mid (c_1, \ldots, c_n) \in q^{\mathcal{I}}$ for all $\mathcal{I}$ s.t. $\mathcal{I} \models \mathcal{T}\}$, where $q^{\mathcal{I}}$ denotes the result of evaluating $q$ in the interpretation $\mathcal{I}$.

We assume that each peer $P$ is only able to provide the certain answers $cert(q, K)$, inferable from its knowledge base $K$ to queries accepted by $P$ itself.

Now ideally we would like, given a client's query $q$ accepted by the peer $P_\ell$, to return all certain answers that are inferable from all the knowledge in the system. That is, we are interested in $cert(q, K_\ell \cup K_r \cup M_\ell)$. To do so we need to exploit the kind of certain answers that peers can compute, i.e., certain answers wrt their knowledge base. We can directly use $cert(q, K_\ell)$ provided by the local peer $P_\ell$, while to use the certain answers provided by the remote peer $P_r$, we need to reformulate the query $q$ into a finite set of queries $\{q_r^1, \ldots, q_r^n\}$ each accepted by the remote peer $P_r$, and require that

$$cert(q, K_\ell \cup K_r \cup M_\ell) = cert(q, K_\ell) \cup \bigcup_{i=1}^{n} cert(q_r^i, K_r) \qquad (1)$$

Formally, we define the *What To Ask* problem, $WTA(q, P_\ell, P_r)$, as follows: *given as input a local peer $P_\ell$, and a query $q$ accepted by $P_\ell$, find a finite set of queries $\{q_r^1, \ldots, q_r^n\}$, each accepted by the remote peer $P_r$, such that condition (1) holds*[3]. This is the problem we tackle in this paper.

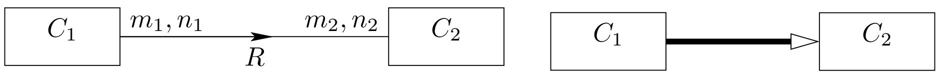
## 3 WTA Problem in an Ontology-Based Framework

We now consider a particular instantiation of the formal framework for knowledge-based peers described above, and provide for such case a solution to the WTA problem.

**Specialized Framework** To specialize our formal framework, we consider specific choices for the language used for specifying peer knowledge bases, queries accepted by peers, and local queries of mapping assertions. We focus first on the language for the peer knowledge base. The language we use, called $L_K^O$ in the sequel, is a subset of FOL that captures the fundamental features of frame-based knowledge representation formalisms and of ontology languages for the Semantic Web. The alphabet of $L_K^O$ consists of constants from $\Gamma$, and of unary and binary predicates, called *classes* and *roles* respectively. Classes denote sets of objects, while roles denote binary relationships between classes. The language $L_K^O$ consists of two components, to represent respectively intensional and extensional knowledge in the peer knowledge base $K$.

The intensional component of $L_K^O$ allows for capturing typical ontology constructs, namely typing of roles, mandatory participation to roles for the objects in a class, functionality of roles, and subsumption between classes. We call the intensional component of $K$ the *schema* of $K$. To keep the presentation simple, we represent the constructs of $L_K^O$ using a graphical notation, and specify their semantics in FOL. Specifically, the schema of $K$ is a directed graph whose nodes are classes and whose edges represent either roles or subsumption relationships. Classes of $K$, in the following denoted by the letter $C$, possibly with subscripts, are represented by means of a rectangle containing the name of the class. Roles of $K$, in the following denoted by the letter $R$, possibly with subscripts, are represented by means of a (thin) arrow, labeled with the name of the role, connecting two classes, called respectively the first and second component of the role. Each role is also labeled with participation and functionality constraints for both components, as depicted in Figure 1 (a), where $m_1$, $m_2$ may be either $0$ (meaning no constraint) or 1 (meaning mandatory participation), and $n_1$, $n_2$ may be either 1 (meaning functionality) or $\infty$ (meaning no constraint). In a schema we omit constraints of the form $(0, \infty)$. The FOL formulas that specify the semantics of the fragment of schema shown in Figure 1 (a) are the following:

[3] Note that in finding $q_r^i$ we can exploit neither $K_r$ nor $V_r$, since $P_r$ is only used as a parameter to the problem for formulating the notion *accepted by* $P_r$.




(a) (b)

**Fig. 1.** A role (a) and a subsumption (b) in the schema of a peer knowledge base

- an assertion that specifies the typing of the two components of the role: $\forall x, y. R(x, y) \supset C_1(x) \wedge C_2(y)$;
- possibly, assertions specifying the mandatory participation to the role:
  - if $m_1 = 1$, then: $\forall x. C_1(x) \supset \exists y. R(x, y)$;
  - if $m_2 = 1$, then: $\forall y. C_2(y) \supset \exists x. R(x, y)$;
- possibly, assertions specifying functionality of the role:
  - if $n_1 = 1$, then: $\forall x, y_1, y_2. R(x, y_1) \wedge R(x, y_2) \supset y_1 = y_2$;
  - if $n_2 = 1$, then: $\forall x_1, x_2, y. R(x_1, y) \wedge R(x_2, y) \supset x_1 = x_2$.

$L_K^O$ is equipped with a subsumption relationship between classes, denoted by a thick hollow arrow from the subsumed class to the subsuming class, as shown in Figure 1 (b). The corresponding FOL formula specifying the semantics is: $\forall x. C_1(x) \supset C_2(x)$.

The extensional component of $L_K^O$ contains facts and existential formulas, possibly involving constants of $\Gamma$. Specifically, each such formula has one of the forms

$$C(a), \quad \exists x. C(x), \quad R(a_1, a_2), \quad \exists x. R(a, x), \quad \exists x. R(x, a), \quad \exists x_1, x_2. R(x_1, x_2),$$

where $C$ and $R$ are respectively a class and a role of the schema of $K$, and $a, a_1, a_2 \in \Gamma$.

As for the language of queries accepted by a peer, we adopt the language of conjunctive queries. A *conjunctive query* (CQ) $q$ is written in the form $\{z_1, \ldots, z_n \mid \exists y_1, \ldots, y_m. \phi(z_1, \ldots, z_n, y_1, \ldots, y_m)\}$, where $z_1, \ldots, z_n$ are (not necessarily pairwise distinct) variables or constants of $\Gamma$, and $\phi(z_1, \ldots, z_n, y_1, \ldots, y_m)$ is a conjunction of atoms, possibly containing constants of $\Gamma$, whose predicates are classes or roles, and whose free variables are the variables in $z_1, \ldots, z_n, y_1, \ldots, y_m$. We call $(z_1, \ldots, z_n)$ the *head* of $q$. Note that a CQ written in the form above corresponds to a FOL query $\{x_1, \ldots, x_n \mid \exists y_1, \ldots, y_m. \phi(x_1, \ldots, x_n, y_1, \ldots, y_m) \wedge \mathit{eqs}\}$, where $x_1, \ldots, x_n$ are pairwise distinct variables, and $\mathit{eqs}$ is a conjunction of equalities, with one equality $x_i = c$ whenever $z_i$ is a constant $c$, and one equality $x_i = x_j$, whenever $z_i$ is the same variable as $z_j$.

The language we adopt to express the local query in a mapping assertion, together with the language of CQs used for the remote query, allows for establishing a basic form of correspondence between knowledge in different peers, namely to map a single element of the knowledge base of the local peer to a CQ over the exported view of a remote peer. Hence, each local query in a mapping assertion is just a single atom (different from equality), and each mapping assertion (in the local peer) has one of the forms $q_r \leadsto \{x \mid C(x)\}$ or $q'_r \leadsto \{x_1, x_2 \mid R(x_1, x_2)\}$, where $q_r$ (resp., $q'_r$) is a CQ over the exported view of the remote peer of arity 1 (resp., 2), and $C$ (resp., $R$) is a concept (resp., role) of the local peer. Moreover, we assume that for each concept $C$ or role $R$ of the local peer, there is at most one mapping assertion in which $C$ (resp., $R$) is used in the local query.

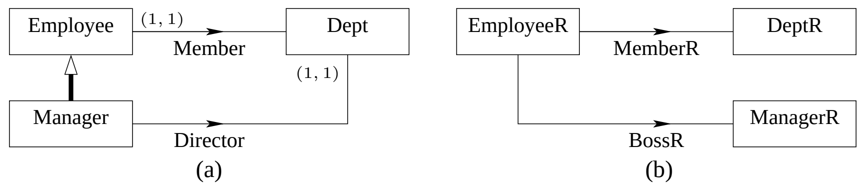



**Fig. 2.** Local and remote schemas for Example 1

*Example 1.* Consider a local peer $P_\ell = \langle K_\ell, V_\ell, M_\ell \rangle$ in which the schema for the local knowledge base $K_\ell$ is the one shown in Figure 2(a), and a remote peer $P_r = \langle K_r, V_r, \emptyset \rangle$ whose schema for $K_r$ is as in Figure 2(b). Assuming that $V_r$ coincides with the set of concepts and roles of the remote schema, a possible set of mapping assertions for the local peer is the following:

$$\{x, y \mid \exists z. BossR(x, z) \wedge MemberR(z, y)\} \leadsto \{x, y \mid Director(x, y)\},$$
$$\{x \mid DeptR(x)\} \leadsto \{x \mid Dept(x)\}, \quad \{x, y \mid MemberR(x, y)\} \leadsto \{x, y \mid Member(x, y)\},$$
$$\{x \mid ManagerR(x)\} \leadsto \{x \mid Manager(x)\}, \ \ \{x \mid EmployeeR(x)\} \leadsto \{x \mid Employee(x)\}.$$

**Computing WTA** We now present an algorithm that solves WTA in the above setting. Let $P_\ell = \langle K_\ell, V_\ell, M_\ell \rangle$ be a local peer, $P_r = \langle K_r, V_r, \emptyset \rangle$ a remote peer, and $q$ a CQ accepted by $P_\ell$. In a nutshell, from the query $q$, the algorithm first produces a set $Q$ of conjunctive queries expressed over $K_\ell$, in which the knowledge of the local ontology that is relevant for answering $q$ has been compiled; then, according to the mapping $M_\ell$, it reformulates the queries in $Q$ in a set of queries that can be accepted by the remote peer. In the following we assume that the theory $K_\ell \cup K_r \cup M_\ell$ is consistent.

Let us now formally describe the algorithm. To this aim, we need some preliminary definitions. Given a CQ $q$, we say that a variable $x$ is *unbound* in $q$ if it occurs only once in $q$, otherwise we say that $x$ is *bound* in $q$. Notice that variables occurring in the head of the query are all bound. A *bound term* is either a bound variable or a constant.

In Figure 3 we define the algorithm compute-WTA. In the algorithm, each unbound variable is represented by the symbol $\_$. Also, $q[g/g']$ denotes the query obtained from $q$ by replacing the atom $g$ with a new atom $g'$.

For each query $q \in Q$, the algorithm first checks if there exists an assertion stating a semantic relation among classes and roles of $K_\ell$ that can be used to produce a new query to be added to the set $Q$ (steps (a) and (b)). Three kinds of assertions are taken into account *(i)* subsumption between classes, *(ii)* participation of classes in roles, *(iii)* mandatory participation of classes in roles. Roughly speaking, atoms in the query $q$ can be reformulated by "navigating" these assertions. In other words, compute-WTA makes use of the assertions in $K_\ell$ as rewriting rules that allow to reformulate the original query $q$ into a set of queries compiling away the knowledge specified by $K_\ell$ that is relevant for computing $cert(q, K_\ell \cup K_r \cup M_\ell)$.

Then, compute-WTA checks if $q$ contains two atoms $g_1$ and $g_2$ that unify (step (c)). In this case, it computes the query *reduce*$(q, g_1, g_2)$, which is obtained by applying to the query $q$ the *most general unifier* between $g_1$ and $g_2$ [11]. This new query is then transformed by the function $\tau$, which replaces with $\_$ each variable symbol $x$ that occurs only once in $q$. The use of $\tau$ is necessary in order to guarantee that each unbound variable is represented by the symbol $\_$. Such a query is then added to $Q$.

**Algorithm** compute-WTA$(q, P_\ell)$
**Input:** CQ $q$, ontology-based peer $P_\ell = \langle K_\ell, V_\ell, M_\ell \rangle$
**Output:** set of conjunctive queries $Mref(Q, M_\ell)$ over $P_r$
$Q \leftarrow \{q\}$;
**repeat**
  $Q_{aux} \leftarrow Q$;
  **for each** $q \in Q_{aux}$ **do**
  (a) **for each** atom $C(x)$ in $q$ **do**
    **for each** assertion in $K_\ell$ stating that a class $D$ is subsumed by the class $C$ **do**
      $Q \leftarrow Q \cup \{ q[C(x)/D(x)] \}$;
    **for each** assertion in $K_\ell$ stating that one of the components of a role $R$ is of type $C$ **do**
      **if** $C$ is the first component of $R$ **then** $Q \leftarrow Q \cup \{ q[C(x)/R(x, \_)] \}$
      **else** $Q \leftarrow Q \cup \{ q[C(x)/R(\_, x)] \}$;
  (b) **for each** atom $R(x, y)$ in $q$ **do**
    **for each** assertion in $K_\ell$ stating the mandatory participation of a class $C$ in the role $R$ **do**
      **if** $C$ is the first component of $R$ and $y$ is _ **then** $Q \leftarrow Q \cup \{ q[R(x, y)/C(x)] \}$;
      **if** $C$ is the second component of $R$ and $x$ is _ **then** $Q \leftarrow Q \cup \{ q[R(x, y)/C(y)] \}$;
  (c) **for each** pair of atoms $g_1, g_2$ in $q$ **do**
    **if** $g_1$ and $g_2$ unify **then** $Q \leftarrow Q \cup \{\tau(\textit{reduce}(q, g_1, g_2))\}$
**until** $Q_{aux} = Q$;
**return** $Mref(Q, M_\ell)$

**Fig. 3.** Algorithm compute-WTA

Finally, compute-WTA reformulates the queries produced in the above steps into a set of queries accepted by the remote peer $P_r$, by means of the procedure $Mref$. Such a procedure implements a standard unfolding technique [9]: roughly speaking, mapping assertions are used as rewriting rules for translating the initial set of queries into a set of queries accepted by the remote ontology.

The following theorem shows correctness and complexity of the algorithm.

**Theorem 1.** *Let $P_\ell = \langle K_\ell, V_\ell, M_\ell \rangle$ be a local peer, $P_r = \langle K_r, V_r, \emptyset \rangle$ a remote peer, and $q$ a CQ accepted by $P_\ell$. Then, compute-WTA$(q, P_\ell)$ returns a solution for $WTA(q, P_\ell, P_r)$ in time polynomial in the size of $K_\ell$ and exponential in the size of $q$.*

*Example 1 (contd.).* Let $q_0 = \{x \mid Employee(x)\}$ be a query accepted by the local peer $P_\ell$, then execute compute-WTA$(q_0, P_\ell)$. Since $Manager$ is subsumed by $Employee$, the algorithm produces the query $q_1 = \{x \mid Manager(x)\}$, and since $Employee$ is the first component of the role $Member$, the algorithm produces $q_2 = \{x \mid Member(x, \_)\}$. No other atom reformulations are generated by the algorithm. Then, compute-WTA applies the operator $Mref$ to the set $Q = \{q_0, q_1, q_2\}$, thus returning the queries $\{x \mid EmployeeR(x)\}$, $\{x \mid ManagerR(x)\}$, $\{x \mid MemberR(x, y)\}$.

## 4 Adding Subsumption between Roles

In this section we show that, if we only add to the ontology language $L_K^O$ the possibility of specifying subsumption relations between roles, WTA may have no solutions.

The definition of $L_K^{O+}$ is the same as the previous ontology language $L_K^O$. In addition, we allow for expressing subsumption relations between pairs of roles. We represent an assertion stating that a role $R_1$ is subsumed by a role $R_2$ by using a thick hollow

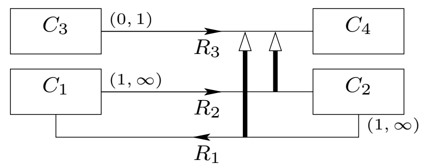



**Fig. 4.** Ontology of peer $P_\ell$ in the proof of Theorem 2

arrow from $R_1$ to $R_2$. The corresponding FOL formula specifying the semantics is: $\forall x, y. R_1(x, y) \supset R_2(x, y)$. We now prove that, for peers using the ontology language $L_K^{O+}$, the problem WTA does not have always a solution.

**Theorem 2.** *There exists a pair of knowledge-based peers $P_\ell, P_r$ with $K_\ell, V_r \in L_K^{O+}$ and a CQ $q$ accepted by $P_\ell$ such that $WTA(q, P_\ell, P_r)$ has no solutions whenever the language accepted by $P_r$ is a subset of the language of FOL queries.*

*Proof.* We exhibit an example of a pair of knowledge-based peers $P_\ell, P_r$ such that the thesis holds. More precisely, let us consider $P_\ell = \langle K_\ell, V_\ell, M_\ell \rangle$, $P_r = \langle K_r, V_r, \emptyset \rangle$ where: $V_\ell$ is the ontology displayed in Figure 4; $K_\ell = V_\ell \cup \{R_1(a, b)\}$; $M_\ell$ consists of the single assertion $\{x, y \mid R_r(x, y)\} \leadsto \{x, y \mid R_3(x, y)\}$; $V_r = \{R_r\}$; $K_r$ is simply a set of facts for $R_r$. We prove that the answer to the boolean query $\{R_1(c, d)\}$ over $V_\ell$ is true if and only if the following condition holds:

*[COND] There exist $n + 1$ constants $a_1, \ldots, a_{n+1}$ (with $n$ even) such that $a_1 = b$, $a_n = c$, $a_{n+1} = d$ and $R_r(a_i, a_{i+1}) \in K_r$ for $1 \leq i \leq n$.*

Indeed, if condition [COND] holds, then, due to the functionality of the participation of $C_3$ to $R_3$, to the two subsumption relations between the three roles, and to the two mandatory participations of $C_1$ and $C_2$, in each model $\mathcal{I}$ for $K_\ell \cup K_r \cup M_\ell$ each tuple of the form $\langle a_i, a_{i+1} \rangle$ must belong to $R_1^{\mathcal{I}}$ if $i$ is even and to $R_2^{\mathcal{I}}$ if $i$ is odd, which implies that $\langle c, d \rangle \in R_1^{\mathcal{I}}$. Conversely, if [COND] does not hold, then it is immediate to exhibit a model $\mathcal{I}$ for $K_\ell \cup K_r \cup M_\ell$ in which the tuple $\langle c, d \rangle$ is not in $R_1^{\mathcal{I}}$. Then, observe that verifying the above condition [COND] requires to compute the transitive closure of $R_r$, which in general cannot be done through a finite number of FOL queries over $P_r$.

The above theorem highlights the crucial role played by the expressiveness of the language for specifying the local peer knowledge base $K_\ell$ in the problem WTA: indeed, by simply adding the possibility of expressing role subsumption to our specialized framework, we miss the property that a solution to the problem WTA always exists, even if we empower the answering abilities of the remote peer to the full FOL language.

## 5 Conclusions

In this paper we have formally defined the What-To-Ask problem, which captures a fundamental issue in a networked environment based on information exchange. We have seen that even small changes in the representation formalism may affect seriously the ability of dealing with this problem. To show this, it has been sufficient to look at a simplified setting with only two interoperating peers. The impact of having more than two peers has been studied in [5], where, however, the peers taken into account are not knowledge bases. Also, query answering in the case where the knowledge bases at the peers are mutually inconsistent (in the line of [2]) remains to be investigated.

## 6 Acknowledgments

This research has been partially supported by the projects INFOMIX (IST-2001-33570), SEWASIE (IST-2001-34825) and INTEROP Network of Excellence (IST-508011) funded by the EU, by the project "Società dell'Informazione" subproject SP1 "Reti Internet: Efficienza, Integrazione e Sicurezza" funded by MIUR – Fondo Speciale per lo Sviluppo della Ricerca di Interesse Strategico, and by project HYPER, funded by IBM through a Shared University Research (SUR) Award grant.

## References

1. P. A. Bernstein, F. Giunchiglia, A. Kementsietsidis, J. Mylopoulos, L. Serafini, and I. Zaihrayeu. Data management for peer-to-peer computing: A vision. In *Proc. of WebDB 2002*, 2002.
2. A. Calì, D. Lembo, and R. Rosati. Query rewriting and answering under constraints in data integration systems. In *Proc. of IJCAI 2003*, pages 16–21, 2003.
3. D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. What to ask to a peer: Ontology-based query reformulation. In *Proc. of KR 2004*, 2004.
4. D. Calvanese, G. De Giacomo, and M. Lenzerini. A framework for ontology integration. In *Proc. of SWWS 2001*, pages 303–316, 2001. Available at `http://www.semanticweb.org/SWWS/program/full/SWWSProceedings.pdf`.
5. D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Logical foundations of peer-to-peer data integration. In *Proc. of PODS 2004*, 2004.
6. A. Halevy, Z. Ives, D. Suciu, and I. Tatarinov. Schema mediation in peer data management systems. In *Proc. of ICDE 2003*, pages 505–516, 2003.
7. J. Heflin and J. Hendler. A portrait of the semantic web in action. *IEEE Intelligent Systems*, 16(2):54–59, 2001.
8. R. Hull, M. Benedikt, V. Christophides, and J. Su. E-services: a look behind the curtain. In *Proc. of PODS 2003*, pages 1–14. ACM Press and Addison Wesley, 2003.
9. M. Lenzerini. Data integration: A theoretical perspective. In *Proc. of PODS 2002*, pages 233–246, 2002.
10. H. J. Levesque and G. Lakemeyer. *The Logic of Knowledge Bases*. The MIT Press, 2001.
11. J. W. Lloyd. *Foundations of Logic Programming (Second, Extended Edition)*. Springer, Berlin, Heidelberg, 1987.
12. N. M. Mattos. Integrating information for on demand computing. In *Proc. of VLDB 2003*, pages 8–14, 2003.
13. M. P. Papazoglou, B. J. Kramer, and J. Yang. Leveraging Web-services and peer-to-peer networks. In *Proc. of CAiSE 2003*, pages 485–501, 2003.
14. L. Serafini and C. Ghidini. Using wrapper agents to answer queries in distributed information systems. In *Proc. of the 1st Int. Conf. on Advances in Information Systems (ADVIS-2000)*, volume 1909 of *LNCS*. Springer, 2000.
15. S. Thakkar, J.-L. Ambite, and C. A. Knoblock. A view integration approach to dynamic composition of Web services. In *Proc. of 2003 ICAPS Workshop on Planning for Web Services*, 2003.
16. Y. Tzitzikas, P. Constantopoulos, and N. Spyratos. Mediators over ontology-based information sources. In *Proc. of WISE 2001*, pages 31–40, 2001.