Principles of peer-to-peer data integration

Maurizio Lenzerini

Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113, I-00198 Roma, Italy
lenzerini@dis.uniroma1.it

Abstract. Integrating heterogeneous computational resources and databases, which are distributed over highly dynamic computer networks, is one of the crucial challenges at the current evolutionary stage of Information Technology infrastructures. Large enterprises, business organizations, e-government systems, and, in short, any kind of internetworking community, need today an integrated and virtualized access to distributed information resources, which grow in number, kind, and complexity. Most of the formal approaches to data integration refer to an architecture based on a global schema and a set of sources. The sources contain the real data, while the global schema provides a reconciled, integrated, and virtual view of the underlying sources. As observed in several contexts, this centralized achitecture is not the best choice for supporting data integration, cooperation and coordination in highly dynamic computer networks. A more appealing architecture is the one based on peer-to-peer systems. In these systems every peer acts as both client and server, and provides part of the overall information available from a distributed environment, without relying on a single global view. In this paper, we study the problem of data integration in peer-to-peer systems, with the aim of singling out the principles that should form the basis for the design of data integration systems in this architecture. Particular emphasis is given to the problem of assigning formal semantics to peer-to-peer data integration. We discuss two different methods for defining such a semantics, and we compare them with respect to the above mentioned principles.

1 Introduction

Integrating heterogeneous computational resources and databases, which are distributed over highly dynamic computer networks, is one of the crucial challenges at the current evolutionary stage of Information Technology infrastructures. Large enterprises, business organizations, e-government systems, and, in short, any kind of internetworking community, need today an integrated and virtualized access to distributed information resources, which grow in number, kind, and complexity. The notion of Virtual Organization denotes a category of modern, Information Technology based establishments, which are able to leverage any available information source, and to interoperate with other parties, efficiently

and with affordable costs, thus benefiting of a significant return of technological investments [6]. The issue of how to support information integration and coordination in Virtual Organization has been addressed in different contexts, including data integration [21], the Semantic Web [16], Peer-to-Peer and Grid computing [2, 14], service oriented computing and distributed agent systems [25, 18].

Most of the formal approaches to data integration refer to an architecture based on a global schema and a set of sources. The sources contain the real data, while the global schema provides a reconciled, integrated, and virtual view of the underlying sources. One of the challenging issues in these systems is to answer queries posed to the global schema. Due to the architecture of the system, query processing requires a reformulation step: the query over the global schema must be re-expressed in terms of a set of queries over the sources [15, 17, 26, 21, 24].

As observed in several contexts, the traditional, centralized achitecture of data integration systems is not the best choice for supporting Virtual Organizations. A more appealing architecture is the one based on peer-to-peer systems. In these systems every peer acts as both client and server, and provides part of the overall information available from a distributed environment, without relying on a single global view. A suitable infrastructure is adopted for managing the information in the various peers.

In this paper, we study the problem of data integration in peer-to-peer systems, with the aim of singling out the principles that should form the basis for the design of data integration systems in this architecture. Differently from the traditional setting, integration in a peer-to-peer architecture is not based on a global, centralized schema. Instead, each peer represents an autonomous information system, and information integration is achieved by establishing peer-to-peer mappings, i.e., mappings among the various peers. We assume that the various peers export data in terms of a suitable schema, and mappings are established among such peer schemas. A peer schema is therefore intended to export the intensional level of information as viewed from the peer. Queries are posed to one peer, and therefore in terms of one peer schema, and the role of query processing is to exploit both the data that are internal to the peer, and the mappings with other peers in the system.

One of the main issues in formalizing peer-to-peer data integration systems is the semantic characterization of peer-to-peer mappings. In this paper, we argue that, although correct from a formal point of view, the usual approach of resorting to a first-order logic interpretation of peer-to-peer mappings (followed e.g. by [7, 14, 2]), has several drawbacks, both from the modeling and from the computational perspective. In particular we analyze three central principles that should form the basis of peer-to-peer data integration:

- *Modularity*: i.e., how autonomous are the various peers in a P2P system with respect to the semantics. Indeed, since each peer is autonomously built and managed, it should be clearly interpretable both alone and when involved in interconnections with other peers. In particular, interconnections with

other peers should not radically change the interpretation of the concepts expressed in the peer.
- *Generality*: i.e., how free we are in placing connections (peer-to-peer mappings) between peers. This is a fundamental property, since actual interconnections among peers are not under the control of any actor in the system.
- *Decidability*: i.e., are sound, complete and terminating query answering mechanisms available? If not, it becomes critical to establish basic quality assurance of the answers returned by the system.

We show that these desirable properties are weakly supported by approaches based directly on first-order logic semantics, and we discuss a recent proposal of a new semantics, aiming at better meeting the above criteria.

The paper is organized as follows. In Section 2, we present a general framework for peer-to-peer data integration. In Section 3, we discuss the semantics of a peer-to-peer data integration system. We distinguish between two types of semantics, one based of traditional first-order logic, and the other based on epistemic logic. In Section 4 we compare the two semantics on the basis of the above mentioned principles. Our main conclusion is that the epsitemic logic semantics allows to overcome important drawbacks of the semantics based on first-order logic. Finally, Section 5 concludes the paper by pointing out several research issues not addressed here.

2 A framework for peer-to-peer data integration

In this section, we present a general framework for peer-to-peer (P2P) data integration. Our goal is to formalize a P2P data integration system as a system constituted by a number of peers, where each peer holds local data, and is connected to other peers by means of declaratively specified mapping assertions. The framework is the one described in [5], and has been inspired by [2, 4, 7, 14]. In the description of the framework, we make use of the following notions.

- We refer to a fixed, infinite, denumerable, set Γ of constants. Such constants are shared by all peers, and are the constants that can appear in the P2P data integration system.
- Given a relational alphabet A, we denote with $\mathcal{L}_A$ the set of function-free first-order logic (FOL) formulas whose relation symbols are in A and whose constants are in Γ.
- A *conjunctive query* (CQ) of arity n over an alphabet A is written in the form

$$\{\mathbf{x} \mid \exists \mathbf{y}\ body_{cq}(\mathbf{x}, \mathbf{y})\}$$

  where $body_{cq}(\mathbf{x}, \mathbf{y})$ is a conjunction of atoms of $\mathcal{L}_A$ involving the free variables (also called the *distinguished* variables of the query) $\mathbf{x} = x_1, \ldots, x_n$, the existentially quantified variables (also called the *non-distinguished* variables of the query) $\mathbf{y} = y_1, \ldots, y_m$, and constants from Γ.

A *P2P data integration system* $\mathcal{P}$ is constituted by a set of peers, each of which includes a specification of the data held by the peer, and a set of mappings that specify the semantic relationships with the data exported by other peers.

Formally, each peer $P \in \mathcal{P}$ is defined as a tuple $P = (G, S, L, M)$, where (cf. [14]):

- G is the *schema* of P, which is a finite set of formulas of $\mathcal{L}_{A_G}$, where A_G is a relational alphabet (disjoint from the other alphabets in $\mathcal{P}$) called the *alphabet* of P. The schema of P is intended to represent the intensional description of the information managed by P, and exported to the other peers.
- S is the *(local) source schema* of P, that is simply a finite relational alphabet (again disjoint from the other alphabets in $\mathcal{P}$), which is called the *local alphabet* of P. The source schema describes the structure of the data sources of the peer (possibly obtained by wrapping physical sources). Such sources are under the control of the peer, and store the real data managed by the peer itself.
- L is a set of *(local) mapping assertions* between G and S. Each local mapping assertion is an expression of the form

  $$cq_S \rightsquigarrow cq_G$$

  where cq_S and cq_G are two conjunctive queries of the same arity, over the source schema S and over the peer schema G, respectively. The local mapping assertions establish the connection between the elements of the source schema and those of the peer schema. In particular, an assertion of the form $cq_S \rightsquigarrow cq_G$ specifies that all the data satisfying the query cq_S over the sources also satisfy the concept in the peer schema represented by the query cq_G. This form of mapping is one of the most expressive among those studied in the data integration literature. Indeed, in terms of the terminology used in data integration, except for the P2P mapping assertions, a peer in our setting corresponds to a GLAV *data integration system* [12, 24] managing a set of sound data sources S defined in terms of a (virtual) global schema G.
- M is a set of *P2P mapping assertions*, each of which is an expression of the form

  $$cq' \rightsquigarrow cq$$

  The query cq, called the *head* of the assertion, is a conjunctive query over the peer (schema of) P, while the query cq', called the *tail* of the assertion, is a conjunctive query of the same arity as cq, over (the schema of) one of the other peers in $\mathcal{P}$. Finally, a P2P mapping assertion $cq' \rightsquigarrow cq$, where cq is a query over the schema of the peer P, expresses the fact that P can use, besides the data in its local sources, also the data retrieved by cq' from the peer P' over (the schema of) which cq' is expressed. Such data are mapped to the schema of P according to what is specified by the query cq. Observe that no limitation is imposed on the topology of the whole set of P2P mapping assertions in the peer system $\mathcal{P}$, and hence the graph corresponding to $\mathcal{P}$ may

be cyclic. The graph corresponding to $\mathcal{P}$ contains one node for every relation symbol in the peer schemas of $\mathcal{P}$, and one edge from the node corresponding to R_1 to the node corresponding to R_2 if there is a P2P mapping assertion in $\mathcal{P}$ whose tail mentions R_1 and whose head mentions R_2.

Finally, we assume that *queries* that are posed to the P2P data integration system $\mathcal{P}$ are in fact posed to one of the peers of $\mathcal{P}$ (say P). Such queries are expressed in a certain relational query language $\mathcal{L}_P$ (e.g., conjunctive queries, Datalog, etc.) over the schema of P. In principle, we make no specific assumption on the query language $\mathcal{L}_P$, except that the peer P can indeed process queries belonging to $\mathcal{L}_P$, and we say that the queries in $\mathcal{L}_P$ are those *accepted by* P.

We end this section with a brief discussion on how the above framework takes into account the basic characteristics of a P2P environment. First, the framework supports dinamicity, simply because there is no single global schema to build and maintain. Mappings can be added to and removed from the systems, and these modifications are smoothly taken into account in the whole system. Modularity is ensured by the fact that new peers can be freely added to and removed from the systems. When a new peer enters the system, the only constraint is that the mappings from other peers to the new peer is registered in the metadata held by the peer. As for generality, it is sufficient to note that we did not impose any constraints to the topology of the peers and their mappings.

3 Formal semantics of P2P data integration systems

In this section we discuss the issue of assigning formal semantics to a P2P data integration system of the form specified in the previous section.

Since we are going to use logic, we should start by specifying the interpretaton domain that we will use in our formalization. Our basic assumption is that the various peers are interpreted over a single fixed infinite domain Δ. We also fix the interpretation of the constants in Γ (cf. previous section) so that:

- each $c \in \Gamma$ denotes an element $d \in \Delta$;
- different constants in Γ denote different elements of Δ;
- each element in Δ is denoted by a constant in Γ.

In other words the constants in Γ act as *standard names* [22]. It follows that Γ is actually isomorphic to Δ, so that we can use (with some abuse of notation) constants in Γ whenever we want to denote domain elements. This is a strong assumption, since it implies that all peers use the same domain and the same vocabolary for denoting the elements of such domain, ie.e, the objects of interest. However, we observe that our framework can be easily extended to take into account more realistic assumptions, such as the ones proposed in [19].

Our definition of the semantics of a P2P data integration system is based on the notion of semantics of one peer. Let us then focus first on the semantics of a single peer $P = (G, S, L, M)$ of the system. We call *peer theory of* P the first-order logic (FOL) theory T_P defined as follows.

– The alphabet of T_P is obtained as union of the alphabet $\mathcal{A}_G$ of G and the alphabet of the local sources S of P.
– The axioms of T_P are the formulas in G plus one formula of the form

$$\forall \mathbf{x}\ (\exists \mathbf{y}\ (body_{cq_S}(\mathbf{x}, \mathbf{y})) \supset \exists \mathbf{z}\ body_{cq_G}(\mathbf{x}, \mathbf{z}))$$

for each local mapping assertion $cq_S \rightsquigarrow cq_G$ in L.

Observe that the P2P mapping assertions of P are not considered in T_P, and that T_P is an "open theory", since for the sources in P, T_P takes into account only the schema S, and not the extension.

We call *local source database* for P, a database D for the source schema S, i.e., a finite relational interpretation of the relation symbols in S. An interpretation $\mathcal{I}$ of T_P is a *model of P based on D* if

1. it is a model of the FOL theory T_P, and
2. for each relational symbol $s \in S$, we have that $s^{\mathcal{I}} = s^D$.

Finally, consider a query q of arity n, expressed in the query language $\mathcal{L}_P$ accepted by P. Given an interpretation $\mathcal{I}$ of T_P, we denote with $q^{\mathcal{I}}$ the set of n-tuples of constants in Γ obtained by evaluating q in $\mathcal{I}$ (viewed as a database over the relations in G), according to the semantics of $\mathcal{L}_P$. We define the *certain answers* $ans(q, P, D)$ to q (accepted by P) based on a local source database D for P, as the set of tuples $\mathbf{t}$ of constants in Γ such that for all models $\mathcal{I}$ of P with respect to D, we have that $\mathbf{t} \in q^{\mathcal{I}}$.

We now turn our attention to assigning a semantics to the whole P2P data integration system. We analyze two different methods for specifying such a semantics.

3.1 FOL semantics

The first approach we discuss is the one followed by [7, 20, 14], called the FOL approach. In this approach, one associates to a P2P data integration system $\mathcal{P}$ a *single* (open) FOL theory $T_{\mathcal{P}}$, obtained as the disjoint union of the various peer theories. Again, P2P mappings are not considered in building $T_{\mathcal{P}}$, but they will play a role in specifying the semantics of the whole system.

By extending the approach used for a single peer, we consider a *source database $\mathcal{D}$ for $\mathcal{P}$*, simply as the (disjoint) union of one local source database D for each peer P in $\mathcal{P}$.

We call *FOL model of $T_{\mathcal{P}}$ based on $\mathcal{D}$* an interpretation $\mathcal{I}$ of the FOL theory $T_{\mathcal{P}}$ such that

– $\mathcal{I}$ is a model of the FOL theory $T_{\mathcal{P}}$, and
– for each relational symbol s of the source schemas in the peers of $\mathcal{P}$, we have that $s^{\mathcal{I}} = s^{\mathcal{D}}$.

We observe that we did not impose any condition on the mapping assertions in the definition of a FOL model of $T_{\mathcal{P}}$ based on $\mathcal{D}$. In order to take into account such assertions, we say that an interpretation $\mathcal{I}$ of the FOL theory $T_{\mathcal{P}}$ is a *FOL model of $\mathcal{P}$ based on $\mathcal{D}$* if

– it is a model of $T_{\mathcal{P}}$ based on $\mathcal{D}$, and
– it is also a model of the formula

$$\forall \mathbf{x}\ (\exists \mathbf{y}\ (body_{cq_1}(\mathbf{x}, \mathbf{y})) \supset \exists \mathbf{z}\ body_{cq_2}(\mathbf{x}, \mathbf{z}))$$

for each P2P mapping assertion $cq_1 \rightsquigarrow cq_2$ in the peers of $\mathcal{P}$.

Finally, given a query q over one peer P among those constituting the whole P2P data integration system $\mathcal{P}$, and given a source database $\mathcal{D}$ for $\mathcal{P}$, we define the *certain answers* $ans_{fol}(q, P, \mathcal{P}, \mathcal{D})$ to q in $\mathcal{P}$ based on $\mathcal{D}$ under FOL semantics, as the set of tuples $\mathbf{t}$ of constants in Γ such that for every FOL model $\mathcal{I}$ of $\mathcal{P}$ based on $\mathcal{D}$, we have that $\mathbf{t} \in q^{\mathcal{I}}$.

3.2 Semantics based on epistemic logic

First-order logic is not the only formal system that can be used as a basis for the semantics of P2P data integration. Indeed, we report here a proposal of a new semantics for P2P data integration systems, based on epistemic logic[1]. The presentation is based on [4, 5]. Notably, the semantics presented here is equivalent to the semantics proposed in [11] for P2P systems.

The epistemic semantics has been defined with the following goals in mind:

– Peers in P2P data integration are to be considered autonomous sites that exchange information. In other words, peers are modules, and the modular structure of the system should be explicitly reflected in the definition of its semantics.
– We do not want to limit a-priori the topology of the mapping assertions among the peers in the system. In particular, we do not want to impose acyclicity of assertions.
– A satisfactory semantic characterization should lead to a setting where query answering is decidable, and possibly, polynomially tractable.

Epistemic logic We briefly remind the basic notions of epistemic logic [22, 10]. In epistemic logic, the language is the one of FOL, except that, besides the usual atoms, one can use another form of atoms, namely $\mathbf{K}\phi$, where ϕ is again a formula. An *epistemic logic theory* is a set of axioms that are formulas in the language of epistemic logic.

The semantics of an epistemic logic theory is based on the notion of epistemic interpretation. We remind the reader that we are referring to a unique interpretation domain Γ. An *epistemic interpretation* $\mathcal{E}$ is a pair $(\mathcal{I}, \mathcal{W})$, where $\mathcal{W}$ is a set of FOL interpretations, and $\mathcal{I} \in \mathcal{W}$. The notion of satisfaction of a formula in an epistemic interpretation $\mathcal{E} = (\mathcal{I}, \mathcal{W})$ is analogous to the one in FOL, with the provision that the interpretation for the atoms is as follows:

[1] Technically we resort to epistemic FOL with standard names, and therefore with a fixed domain, and rigid interpretation of constants [22].

– a FOL formula constituted by an atom $a(\mathbf{x})$ (where $\mathbf{x}$ are the free variables in the formula) is satisfied in $(\mathcal{I}, \mathcal{W})$ by the tuples $\mathbf{t}$ of constants in Γ such that $a(\mathbf{t})$ is true in $\mathcal{I}$,
– an atom of the form $\mathbf{K}\phi(\mathbf{x})$ is satisfied in $(\mathcal{I}, \mathcal{W})$ by the tuples $\mathbf{t}$ of constants in Γ such that $\phi(\mathbf{t})$ is satisfied in all epistemic interpretations $(\mathcal{J}, \mathcal{W})$ with $\mathcal{J} \in \mathcal{W}$.

Note that our definition of epistemic interpretation is a simplified view of a Kripke structure of an S5 modal system, in which every epistemic interpretation is constituted by a set of worlds, each one connected, through the accessibility relation, to all the other ones. Indeed, in our setting each world corresponds to a FOL interpretation, and the accessibility relation is left implicit by viewing the whole structure as a set.

An *epistemic model* of an epistemic logic theory is an epistemic interpretation that satisfies every axiom of the theory. In turn, an axiom constituted by the formula ϕ is satisfied by an epistemic interpretation $(\mathcal{I}, \mathcal{W})$ if, for every $\mathcal{J} \in \mathcal{W}$, the epistemic interpretation $(\mathcal{J}, \mathcal{W})$ satisfies ϕ. Observe that in order for an epistemic interpretation $(\mathcal{I}, \mathcal{W})$ to be a model of a theory, the axioms of the theory are required to be satisfied in every $\mathcal{J} \in \mathcal{W}$. Hence, with regard to the satisfaction of axioms, only $\mathcal{W}$ counts.

Observe that, in epistemic logic, the formula $\mathbf{K}(\phi \vee \psi)$ has an entirely different meaning with respect to the formula $\mathbf{K}\phi \vee \mathbf{K}\psi$. Indeed, the former is satisfied in an interpretation $(\mathcal{J}, \mathcal{W})$ if for every $\mathcal{I} \in \mathcal{W}$, there is at least one among $\{\phi, \psi\}$, that is satisfied in $\mathcal{I}$. Conversely, the latter requires either that ϕ is satisfied in all $\mathcal{I} \in \mathcal{W}$ or that ψ is satisfied in all $\mathcal{I} \in \mathcal{W}$. Observe also that, if ϕ is a FOL formula, there is a striking difference between $\mathbf{K}\exists x\phi(x)$ and $\exists x\mathbf{K}\phi(x)$. In particular, for $\exists x\mathbf{K}\phi(x)$ to be satisfied in $(\mathcal{I}, \mathcal{W})$ there must be a constant $c \in \Gamma$ such that $\phi(c)$ is satisfied in every $\mathcal{J} \in \mathcal{W}$, while for $\mathbf{K}(\exists x\phi(x))$ to be satisfied it is only required that in each $\mathcal{J} \in \mathcal{W}$ there exists a constant $c \in \Gamma$ such that $\phi(c)$ is satisfied in $\mathcal{J}$.

Formalizing P2P mapping assertions in epistemic logic We formalize a P2P data integration system $\mathcal{P}$ in terms of the epistemic logic as follows.

As before, we consider the FOL theory $T_{\mathcal{P}}$, obtained as the disjoint union of the various peer theories. To such a theory we add a set of axioms $M_{\mathcal{P}}$ to capture the mapping assertions. In particular, $M_{\mathcal{P}}$ is formed by one axiom of the form

$$\forall \mathbf{x}\ (\mathbf{K}(\exists \mathbf{y}\ (body_{cq_1}(\mathbf{x}, \mathbf{y}))) \supset \exists \mathbf{z}\ body_{cq_2}(\mathbf{x}, \mathbf{z}))$$

for each P2P mapping assertion $cq_1 \rightsquigarrow cq_2$ in the peers of $\mathcal{P}$. These formulas say that for each P2P mapping assertion $cq_1 \rightsquigarrow cq_2$ for every tuple $\mathbf{t}$ of objects in Γ, the fact that $\exists \mathbf{y}\ body_{cq_1}(\mathbf{t}, \mathbf{y})$ is satisfied in every FOL model in $\mathcal{W}$ implies that also $\exists \mathbf{z}\ body_{cq_2}(\mathbf{t}, \mathbf{z})$ is satisfied in every FOL model in $\mathcal{W}$. Note that the formalization of the P2P mapping assertions in terms of the formulas specified above intuitively reflects the idea that only what is *known* by the peers mentioned in the tail of the assertion is transferred to the peer mentioned in the head.

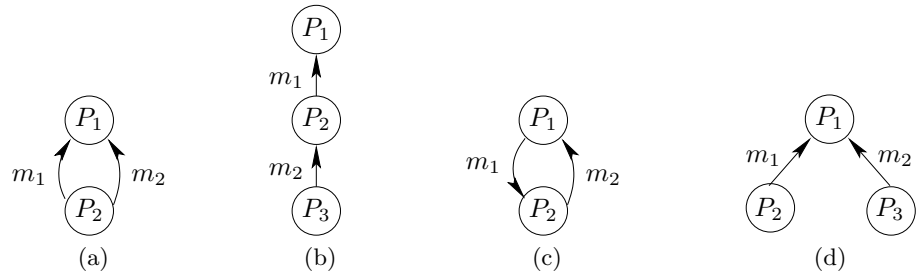

Fig. 1. Interactions between two mappings

We use the notion of FOL model of $T_{\mathcal{P}}$ based on a source database $\mathcal{D}$ for $\mathcal{P}$, as defined above, namely, a FOL model of $T_{\mathcal{P}}$ based on $\mathcal{D}$ is an interpretation $\mathcal{I}$ of the FOL theory $T_{\mathcal{P}}$ such that

- $\mathcal{I}$ is a model of the FOL theory $T_{\mathcal{P}}$, and
- for each relational symbol s of the source schemas in the peers of $\mathcal{P}$, we have that $s^{\mathcal{I}} = s^{\mathcal{D}}$.

Finally, we call *epistemic model of* $\mathcal{P}$ *based on* $\mathcal{D}$ an epistemic interpretation $(\mathcal{I}, \mathcal{W})$ such that

- $\mathcal{W}$ is a set of models of $T_{\mathcal{P}}$ based on $\mathcal{D}$, and
- $(\mathcal{I}, \mathcal{W})$ is an epistemic model of $M_{\mathcal{P}}$.

Now, given a query q over the peer P in $\mathcal{P}$, and given a source database $\mathcal{D}$ for $\mathcal{P}$, we define the *certain answers* $ans_{\mathbf{k}}(q, \mathcal{P}, \mathcal{D})$ to q in $\mathcal{P}$ based on $\mathcal{D}$ under the epistemic semantics, as the set of tuples $\mathbf{t}$ of constants in Γ such that for every epistemic model $(\mathcal{I}, \mathcal{W})$ of $\mathcal{P}$ based on $\mathcal{D}_{\mathcal{P}}$, we have that $\mathbf{t} \in q^{\mathcal{I}}$.

Observe that the epistemic semantics can be considered as a well-behaved, sound approximation of the first-order semantics, since it is immediate to verify that, for each q, $\mathcal{P}$, and $\mathcal{D}$, if $\mathbf{t} \in ans_{\mathbf{k}}(q, \mathcal{P}, \mathcal{D})$, then $\mathbf{t} \in ans_{fol}(q, \mathcal{P}, \mathcal{D})$.

Notably, the semantics based on epistemic logic is the one at the basis of Hyper[2], a joint project carried out by the University of Roma "La Sapienza" and IBM, whose aim is to build a P2P data integration infrastructure based on Data Grids.

4 Comparison between the FOL semantics and the epistemic logic semantics

In this section, we compare the two semantics of P2P data integration systems presented in the previous section, in particular with respect to the three principles mentioned in the introduction, namely modularity, generality, and decidability of query answering.

[2] See `http://www.dis.uniroma1.it/~lenzerini/progetti/hyper`

To highlight the differences between the two semantics, we will consider the simplest setting in which interactions btween various P2P mappings may occur, namely P2P data integration systems containing only two P2P mappings. The three types of systems we discuss in the following are depicted in Figure 1, and represent respectively the case of parallel, sequential, and cyclic composition, where each circle represents a peer, and an arrow from a peer P' to a peer P represents a mapping assertion whose head is a CQ over P and whose tail is a CQ over P'.

We first need to provide some definitions. Given a peer $P = (G, S, L, M)$, we denote as $\tau(P)$ the peer (G, S', L', M) such that:

1. S' is obtained from S by adding a new source predicate symbol r, of the same arity as cq', for each P2P mapping assertion $cq' \rightsquigarrow cq$ in M between a peer P' and P. We also denote as $Q(r)$ the query cq' in the tail of the corresponding P2P mapping assertion, and denote as $P(r)$ the peer P', i.e., the peer over which the query $Q(r)$ is expressed.
2. L' is obtained from L by adding the local mapping assertion $\{\mathbf{x} \mid r(\mathbf{x})\} \rightsquigarrow cq$ for each P2P mapping assertion $cq' \rightsquigarrow cq$ in M.

Furthermore, for a P2P data integration system $\mathcal{P}$, we denote as $\tau(\mathcal{P})$ the P2P data integration system $\{\tau(P) \mid P \in \mathcal{P}\}$. For each peer P, we call *auxiliary alphabet* of P, denoted as $AuxAlph(P)$, the set of new source predicate symbols thus defined. Informally, in each peer the additional sources corresponding to the predicates in the auxiliary alphabet are used to "simulate" the effect of the P2P mapping assertions with respect to contributing to the data of the peer.

4.1 Parallel composition

Let us consider a P2P data integration system $\mathcal{P}_{par}$ with the structure depicted in Figure 1(a). To highlight the interdependence between mappings, we assume that P_1 does not contain local sources (and local mappings). Hence, $\mathcal{P}_{par}$ is constituted by two peers $P_1 = (G_1, \emptyset, \emptyset, \{m_1, m_2\})$, and $P_2 = (G_2, S_2, L_2, \emptyset)$.

Informally, in the context of parallel composition, we can consider a semantics for P2P data integration systems as modular, if for every query q over P_1, and for every source database D_2 for P_2, the certain answers to q in $\mathcal{P}_{par}$ with respect to D_2 under the considered semantics can be computed by first populating P_1 with the data retrieved by independently applying the two mappings and then evaluating q over such data. Formally, let m_1 be $cq'_1 \rightsquigarrow cq_1$, let m_2 be $cq'_2 \rightsquigarrow cq_2$, and consider the peer $\tau(P_1) = (G_1, \{r_1, r_2\}, \{m'_1, m'_2\}, \{m_1, m_2\})$, where m'_1 is $\{\mathbf{x} \mid r_1(\mathbf{x})\} \rightsquigarrow cq_1$ and m'_2 is $\{\mathbf{x} \mid r_2(\mathbf{x})\} \rightsquigarrow cq_2$. For a local source database D_2 for P_2, let $\delta(P_1, D_2)$ be the local source database for $\tau(P_1)$ such that $r_1^{\delta(P_1,D_2)}$ coincides with the certain answers $ans(cq'_1, P_2, D_2)$ over the single peer P_2, and $r_2^{\delta(P_1,D_2)}$ coincides with the certain answers $ans(cq'_2, P_2, D_2)$ over P_2. Now, semantics X is modular if for every query q to P_1 and for every source database D_2 for P_2, we have that $ans_X(q, P_1, \mathcal{P}, \{D_2\})$ coincides with the certain answers $ans(q, \tau(P_1), \delta(P_1, D_2))$ over $\tau(P_1)$.

In [5], it is shown that a P2P data integration system as simple as $\mathcal{P}_{par}$ is sufficient to separate the epistemic and the FOL semantics with respect to modularity. In particular, the paper reports the following two results:

- There is a P2P data integration system $\mathcal{P}_{par} = \{P_1, P_2\}$ of the form as above, a source database D_2 for P_2, and a query q to P_1 such that $ans_{fol}(q, P_1, \mathcal{P}, \{D_2\}) \neq ans(q, \tau(P_1), \delta(P_1, D_2))$.
- Let $\mathcal{P}_{par}$ and D_2 be as above. Then, for every query q over P_1 we have that $ans_{\mathbf{k}}(q, P_1, \mathcal{P}, \{D_2\}) = ans(q, \tau(P_1), \delta(P_1, D_2))$.

4.2 Sequential composition

We consider a P2P data integration system $\mathcal{P}_{seq}$ with the structure depicted in Figure 1(b). Again, to highlight the interaction between the mappings, we assume that both P_1 and P_2 do not contain local sources. Hence, $\mathcal{P}_{seq}$ is constituted by three peers $P_1 = (G_1, \emptyset, \emptyset, \{m_1\})$, $P_2 = (G_2, \emptyset, \emptyset, \{m_2\})$, and $P_3 = (G_3, S_3, L_3, \emptyset)$.

Informally, in the context of sequential composition, we can consider a semantics for P2P data integration systems as modular, if for every query q_1 over P_1, and for every source database D_3 for P_3, the certain answers to q in $\mathcal{P}_{seq}$ with respect to D_3 under the considered semantics can be computed by (i) populating P_2 with the data retrieved by applying the mapping m_2, (ii) using such data to populate P_1 by applying the mapping m_1, and (iii) evaluating q over P_1. Formally, let m_1 be $cq_2 \rightsquigarrow cq_1$, let m_2 be $cq_3 \rightsquigarrow cq_2'$, and consider the peers $\tau(P_1) = (G_1, \{r_1\}, \{m_1'\}, \{m_1\})$ with $m_1' = \{\mathbf{x} \mid r_1(\mathbf{x})\} \rightsquigarrow cq_1$ and $\tau(P_2) = (G_2, \{r_2\}, \{m_2'\}, \{m_2\})$ with $m_2' = \{\mathbf{x} \mid r_2(\mathbf{x})\} \rightsquigarrow cq_2'$. For a local source database D_3 for P_3, let $\delta(P_2, D_3)$ be the local source database for $\tau(P_2)$ such that $r_2^{\delta(P_2, D_3)} = ans(cq_3, P_3, D_3)$ and let $\delta(P_1, P_2, D_3)$ be the local source database for $\tau(P_1)$ such that $r_1^{\delta(P_1, P_2, D_3)} = ans(cq_2, P_2, \delta(P_2, D_3))$. Now, semantics X is modular if for every query q to P_1 and for every source database D_3 for P_3, we have that $ans_X(q, P_1, \mathcal{P}, \{D_3\}) = ans(q, \tau(P_1), \delta(P_1, P_2, D_3))$.

In [5], it is shown that, also in the context of sequential composition, while the epistemic semantics for P2P data integration systems is modular, the FOL semantics is not so. In particular,

- There is a P2P data integration system $\mathcal{P}_{seq} = \{P_1, P_2, P_3\}$ of the form as above, a source database D_3 for P_3, and a query q over P_1 such that $ans_{fol}(q, P_1, \mathcal{P}, \{D_3\}) \neq ans(q, \tau(P_1), \delta(P_1, P_2, D_3))$.
- Let $\mathcal{P}_{seq}$ and D_3 be as above. Then, for every query q over P_1 we have that $ans_{\mathbf{k}}(q, P_1, \mathcal{P}, \{D_3\}) = ans(q, \tau(P_1), \delta(P_1, P_2, D_3))$.

A problem related to the one considered here for sequential P2P data integration systems is the one of mapping composition, as defined in [23]. In that paper, the authors study a system in which peer schemas are empty, and P2P mappings are as here (i.e., GLAV mappings between CQs), but interpreted according to the FOL semantics. The authors show that in this setting the composition of two (sets of) P2P mappings is quite involved, and in general is formed by an infinite number of P2P mappings between the first and the last peer.

Interestingly, an immediate consequence of the results in the next section is that, in the epistemic semantics instead, the composition of two (sets of) P2P mappings is formed by a finite set of P2P mappings between the first and the last peer.

4.3 Simple cycle between two peers

Consider a P2P data integration system $\mathcal{P}_{cyc}$ with the structure depicted in Figure 1(c). In [5], it is shown that the presence of a cycle between two peers suffices to make query answering undecidable under the FOL semantics.

Notice that, since P_1 and P_2 are in general designed independently of each other, even if care is taken to retain decidability of query answering for each of them separately, when interconnected in a P2P data integration system, under the FOL semantics there is no way to ensure decidability of query answering in the whole system, since no single actor has the control on all the P2P mappings. This is a further indication of the lack of modularity in systems based on the FOL semantics. Observe also that the only way to retain decidability would be to trade it with generality, by restricting the topology of the P2P mappings [14, 20, 9]. In practice this may even be unfeasible, again since no actor is in control of all P2P mappings.

On the other hand, [4, 5] show that under the epistemic semantics, we can retain both generality and decidability for P2P data integration systems with arbitrary structure. More precisely, [5] presents a distributed algorithm that, given a query q over a peer P in $\mathcal{P}$, and given a source database $\mathcal{D}$ for $\mathcal{P}$, returns the set of certain answers $ans_{\mathbf{k}}(q, \mathcal{P}, \mathcal{D})$ to q in $\mathcal{P}$ based on $\mathcal{D}$ under the epistemic semantics. The algorithm assumes that each peer in the system is able to compute the perfect rewriting of a query with respect to the set of mappings relevant for the peer. Under this hypothesis, the algorithm computes the certain answers in polynomial time with respect to the size of the source database (i.e., in data complexity).

4.4 Data integration

Finally, we consider a P2P system $\mathcal{P}_{di}$ with the structure depicted in Figure 1(d), and we consider the case where P_1 has no local sources, and each of the peers P_2 and P_3 consists of a single data source, i.e., G consists of a single relation and L maps such a relation to the source. This case corresponds to the typical data integration setting, where P_1 acts as the global schema, P_2 and P_3 as sources, and the P2P mappings of P_1 as GLAV mappings between the global schema and the sources. Interestingly, in this case the two semantics coincide. This indicates that the data integration setting does not contain sufficient structure to get into the subtleties that arise in P2P systems. And this justifies why, in data integration, it has not been necessary to introduce semantics based on epistemic notions.

5 Conclusions

In this paper we have discussed basic principles for P2P data integration systems, and we have presented a general framework for such systems. We have also discussed possible methods for specifying the semantics of such systems. Motivated by several drawbacks in the usual FOL formalization of data integration, we have reported on a new semantics proposed in [4, 5], arguing that it is superior with respect to all the principles.

Data integration in P2P system is still in its infancy. Several aspects not addressed in this paper are both interesting from a research point of view, and important from a practical point of view. In particular:

- Peer schemas in the P2P data integration systems considered in this paper are specified just in terms of an alphabet. Obviously, more expressive forms of schema may be needed in real settings.
- According to our framework, a P2P data integration system is based on a set of mappings between peers. Mappings are established between the various peer schemas. Defining such mappings is difficult and time consuming. It is thus essential to design (semi)automatic techniques for deriving and mantaining mappings among peers [8].
- The effectiveness and the efficiency of algorithms for query answering in P2P data integration systems, including the one referred to in this paper, should be tested in realistic settings.
- In our formal framework we assumed the existence of a single, common set of constants for denoting the interpretation domain of all the peers. In real applications, this is a too strong assumption, as the various peers are obviously autonomous in choosing the mechanisms for denoting the domain elements. The issue of different vocabularies of constants in different peers is addressed, for example, in [2, 19].
- Finally, in our current formalization, if the information that one peer provides to another peer is inconsistent with the information known by the latter, the whole P2P data integration system is logically inconsistent. Again, this is a strong limitation when one wants to use the framework in real applications. Data reconciliation and cleaning techniques may mitigate such a problem in some cases. More generally, to deal with this problem, suitable extensions of the epistemic semantics presented here should be investigated, e.g., in the line of [3].

6 Acknowledgments

I warmly thank Diego Calvanese, Giuseppe De Giacomo and Riccardo Rosati, with whom I carried out most of my research work in peer-to-peer data integration.

This research has been partially supported by Projects INFOMIX (IST-2001-33570) and SEWASIE (IST-2001-34825) funded by the EU, by MIUR — Fondo Speciale per lo Sviluppo della Ricerca di Interesse Strategico — project “Società

dell'Informazione", subproject SP1 "Reti Internet: Efficienza, Integrazione e Sicurezza", and by project HYPER, funded by IBM through a Shared University Research (SUR) Award grant.

References

1. K. Aberer, M. Punceva, M. Hauswirth, and R. Schmidt. Improving data access in P2P systems. *IEEE Internet Computing*, 2002.
2. P. A. Bernstein, F. Giunchiglia, A. Kementsietsidis, J. Mylopoulos, L. Serafini, and I. Zaihrayeu. Data management for peer-to-peer computing: A vision. In *Proc. of the 5th Int. Workshop on the Web and Databases (WebDB 2002)*, 2002.
3. A. Calì, D. Lembo, and R. Rosati. Query rewriting and answering under constraints in data integration systems. In *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI 2003)*, 2003. To appear.
4. D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Data integration in p2p systems. In *Databases, Information Systems, and Peer-to-Peer Computing*, pages 77–90. Springer, LNCS 2944, 2004.
5. D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Logical foundations of peer-to-peer data integration. 2004. To appear.
6. L. Camarinha-Matos, H. Afsarmanesh, C. Garita, and C. Lima. Towards an architecture for virtual enterprises. *J. Intelligent Manufacturing*, 9(2), 1998.
7. T. Catarci and M. Lenzerini. Representing and using interschema knowledge in cooperative information systems. *J. of Intelligent and Cooperative Information Systems*, 2(4):375–398, 1993.
8. A. Doan, P. Domingos, and A. Halevy. Learning to match the schemas of data sources: a multistrategy approach. *Machine Learning Journal*, 2003.
9. R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa. Data exchange: Semantics and query answering. In *Proc. of the 9th Int. Conf. on Database Theory (ICDT 2003)*, pages 207–224, 2003.
10. M. Fitting. Basic modal logic. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 1, pages 365–448. Oxford Science Publications, 1993.
11. E. Franconi, G. M. Kuper, A. Lopatenko, and L. Serafini. A robust logical and computational characterisation of peer-to-peer database systems. In *Databases, Information Systems, and Peer-to-Peer Computing*, pages 64–76. Springer, LNCS 2944, 2004.
12. M. Friedman, A. Levy, and T. Millstein. Navigational plans for data integration. In *Proc. of the 16th Nat. Conf. on Artificial Intelligence (AAAI'99)*, pages 67–73. AAAI Press/The MIT Press, 1999.
13. S. Gribble, A. Halevy, Z. Ives, M. Rodrig, and D. Suciu. What can databases do for peer-to-peer? In *Proc. of the 4th Int. Workshop on the Web and Databases (WebDB 2001)*, 2001.
14. A. Halevy, Z. Ives, D. Suciu, and I. Tatarinov. Schema mediation in peer data management systems. In *Proc. of the 19th IEEE Int. Conf. on Data Engineering (ICDE 2003)*, 2003.
15. A. Y. Halevy. Answering queries using views: A survey. *Very Large Database J.*, 10(4):270–294, 2001.
16. J. Heflin and J. Hendler. A portrait of the semantic web in action. *IEEE Intelligent Systems*, 16(2):54–59, 2001.

17. R. Hull. Managing semantic heterogeneity in databases: A theoretical perspective. In *Proc. of the 16th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS'97)*, pages 51–61, 1997.
18. R. Hull, M. Benedikt, V. Christophides, and J. Su. E-services: a look behind the curtain. In *Proc. of the 22nd ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS 2003)*, pages 1–14. ACM Press and Addison Wesley, 2003.
19. A. Kementsietsidis, M. Arenas, and R. J. Miller. Mapping data in peer-to-peer systems: Semantics and algorithmic issues. pages 325–336. ACM Press and Addison Wesley, 2003.
20. C. Koch. Query rewriting with symmetric constraints. In *Proc. of the 2nd Int. Symp. on Foundations of Information and Knowledge Systems (FoIKS 2002)*, volume 2284 of *Lecture Notes in Computer Science*, pages 130–147. Springer, 2002.
21. M. Lenzerini. Data integration: A theoretical perspective. In *Proc. of the 21st ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS 2002)*, pages 233–246, 2002.
22. H. J. Levesque and G. Lakemeyer. *The Logic of Knowledge Bases*. The MIT Press, 2001.
23. J. Madhavan and A. Y. Halevy. Composing mappings among data sources. In *Proc. of the 29th Int. Conf. on Very Large Data Bases (VLDB 2003)*, pages 572–583, 2003.
24. P. McBrien and A. Poulovassilis. Defining peer-to-peer data integration using both as view rules. In *Databases, Information Systems, and Peer-to-Peer Computing*, pages 91–107. Springer, LNCS 2944, 2004.
25. M. P. Papazoglou, B. J. Kramer, and J. Yang. Leveraging Web-services and peer-to-peer networks. In *Proc. of the 15th Int. Conf. on Advanced Information Systems Engineering (CAiSE 2003)*, pages 485–501, 2003.
26. J. D. Ullman. Information integration using logical views. In *Proc. of the 6th Int. Conf. on Database Theory (ICDT'97)*, volume 1186 of *Lecture Notes in Computer Science*, pages 19–40. Springer, 1997.