Book Title
Book Editors
IOS Press, 2003

Data Management in Peer-to-Peer Data Integration Systems

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

[a] *Faculty of Computer Science, Free University of Bozen-Bolzano*
[b] *Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza"*

Abstract. Decentralized data management has been addressed during the years by means of several technical solutions, ranging from distributed DBMSs, to mediator-based data integration systems. Recently, such an issue has been investigated in the context of Peer-to-Peer (P2P) architectures. In this chapter we focus on P2P data integration systems, which are characterized by various autonomous peers, each peer being essentially an autonomous information system that holds data and is linked to other peers by means of P2P mappings. P2P data integration does not rely on the notion of global schema, as in traditional mediator-based data integration. Rather, it computes answers to users' queries, posed to any peer of the system, on the basis of both local data and the P2P mappings, thus overcoming the main drawbacks of centralized mediator-based data integration systems and providing the foundations of effective data management in virtual organizations.

In this chapter we first survey the most significant approaches proposed in the literature for both mediator-based data integration and P2P data management. Then, we focus on advanced schema-based P2P systems for which the aim is semantic integration of data, and analyze the commonly adopted approach of interpreting such systems using a first-order semantics. We show some weaknesses of this approach, and compare it with an alternative approach, based on multi-modal epistemic semantics, which reflects the idea that each peer is conceived as a rational agent that exchanges knowledge/belief with other peers. We consider several central properties of P2P data integration systems: modularity, generality, and decidability. We argue that the approach based on epistemic logic is superior with respect to all the above properties.

Keywords. Peer-to-Peer Data Integration, Semantics, Algorithms

1. Introduction

Data management systems have been continuously evolving during the years to respond to customer demand and the new market requirements. Starting from the late 80s, centralized systems, which had often produced huge, monolithic, and generally inefficient databases, have been replaced by decentralized systems in which data are maintained in different sites with autonomous storage and computation capabilities. All such systems

[1] Correspondence to: Domenico Lembo, Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Via Salaria 113, 00198 Roma, Italy. Tel.: +39 06 4991 8339; Fax: +39 06 8530 0849; E-mail: lembo@dis.uniroma1.it.

are characterized by an architecture in which data returned to a user query might not be physically stored at the site queried by the user. In distributed databases, decentralization of data is generally achieved to enhance system performance, and is precisely designed and controlled. However, such an architecture is not able to support the integration of previously existing systems, where data dispersed over several sources are required to be accessed in a centralized and uniform way. Database federation tools enable data from multiple heterogeneous data sources to appear as if it was contained in a single federated database. Such tools provide mechanisms which mask the native characteristics of each source and represent it in a common format, thus enabling a centralized and transparent data access. Mediator-based data integration systems (schematized in Figure 1) provide in addition the capability of defining a (virtual) global schema representing the unified view of the application domain, which is related to the sources through a suitable mapping establishing a semantic relationship between them. Here, the integration can be performed in a declarative way, and query answering, i.e., the problem of providing answers to users' queries posed on the virtual global schema, is in general a form of reasoning with incomplete information, and is achieved by means of powerful mechanisms and advanced techniques. More recently, the issue of cooperation, integration, and coordination

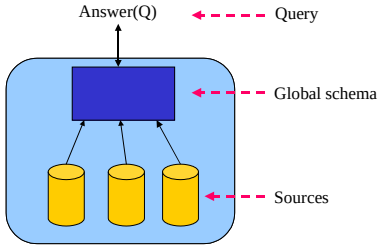

Figure 1. Mediator-based Data Integration System

between data nodes in open distributed systems has been investigated in the context of Peer-to-Peer (P2P) data management [50]. In short, a P2P system is characterized by a structure constituted by various autonomous nodes (called sources, sites, agents, or peers, depending on the context in which such systems are studied) that hold data and that are linked to other nodes by means of mappings. Differently from all the above mentioned architectures, P2P systems do not require a centralized management and are not developed under the control of central authority. Each peer provides part of the overall information available from a distributed environment, and acts both as a client and as a server in the system. The result is a completely decentralized architecture, flexible and able to handle dynamic changes in the system, which peers can join or leave at run-time. Then, favored by its characteristics, P2P computing is expected to soon penetrate the world of information technology, leveraging the growth of virtual organizations willing to share information on the network, as well as supporting the electronic business. As for this last kind of applications, new forms of electronic brokerage are emerging, where each broker is a peer offering goods or services either directly on behalf of a producer, or through another broker, i.e., through a peer to which a percentage fee is due in case of a transaction. In many cases, for example, in the electronic commerce for the tourism domain, such P2P applications are very data intensive as each peer must store large amounts of travel and hotel data or query such data across a heterogeneous P2P network. Furthermore, they

are characterized by a potentially huge number of peers willing to access to such a business. At the same time, mobile end consumer devices such as cell phones and PDAs are becoming more powerful as their processing power and data storage capacities approach the speed and memory of workstations. This opens a perspective of data intensive P2P applications for participants in mobile networks.

Nowadays, apart from the basic structure and algorithms for P2P information integration systems, research and technology on advanced data integration and exchange is at an embryonic state, and an in-depth investigation on the field is still needed in order to achieve powerful, human level P2P data integration. Recent research is devoted to provide techniques for evolving from basic P2P networks supporting only file exchanges using simple filenames as metadata [32,68], to more complex systems like schema-based P2P networks [50,6,37,20]. In particular, in a *P2P data integration system* [50] each peer is essentially a mediator-based data integration system, i.e., it manages a set of local data sources semantically connected, via a *local mapping*, to a (virtual) global schema called the *peer schema*. In addition, the specification of a peer includes a set of *P2P mappings* that specify the relationships with the data exported by other peers, as shown in Figure 2. Information in such systems can be *queried* to any peer (by external users or other peers). The queried peer, by exploiting its P2P mappings, can make use of the data in the other peers for providing the answer.

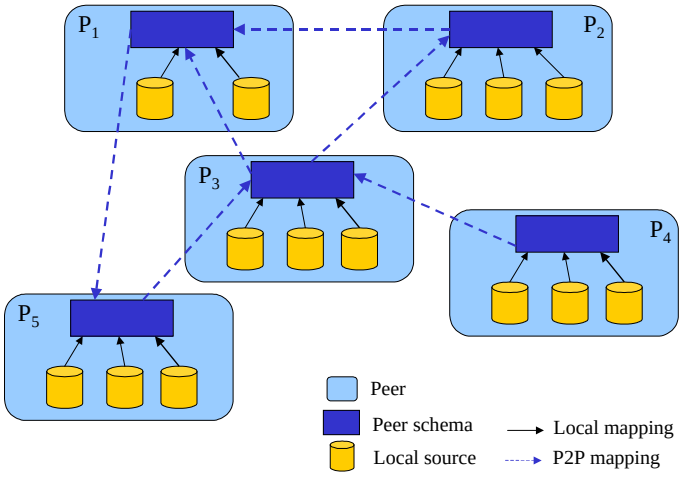

Figure 2. Peer-to-Peer Data Integration System

In this chapter we first survey the most significant approaches proposed in the literature for both mediator-based data integration and P2P data management. Then, we focus on advanced schema-based P2P systems for which the aim is semantic integration of data, and analyze the commonly adopted approach of interpreting such systems using a first-order semantics.

In the following, we survey the most significant approaches proposed in the literature for both mediator-based data integration and P2P data management. In particular, our analysis on P2P data management ranges from first systems developed for content sharing in a networking environment, to advanced schema-based P2P data integration systems. Then, focusing on P2P data integration, we analyze the commonly adopted ap-

proach for interpreting P2P systems using a first-order (FOL) semantics. We show some weaknesses of this approach, and compare it with an alternative approach, based on epistemic semantics. We consider several central properties of P2P data integration systems: modularity, generality, and decidability. We show that the approach based on epistemic logic is clearly superior to the usual approaches based on first-order logic with respect to all the above properties. In particular, we show that, in systems in which peers have decidable schemas and conjunctive mappings, but are arbitrarily interconnected, possibly presenting cycles in the network of peers, the first-order approach may lead to undecidability of query answering, while the epistemic approach always preserves decidability. This is a fundamental property, since the actual interconnections among peers are not under the control of any actor in the system. In this respect, our formalization nicely models the modularity of P2P architectures, i.e., the fact that each peer is autonomous, without resorting to any assumptions such as acyclicity, on the topology of the P2P systems. To this aim, we formalize a P2P data integration system in terms of the multi-modal epistemic logic $\mathsf{S5}_n$, according to which each peer is modeled as a rational agent that exchanges knowledge/belief with other peers. This is in line with the idea of modeling a distributed information system in terms of multi-agent modal logic [34].

The rest of this chapter is organized as follows. In Section 2 we review approaches to both mediator-based data integration and P2P data management. In Section 3 we provide a formal framework for P2P data integration, and in Section 4 we describe classical FOL semantics for interpreting such a framework. Then, in Section 5 we discuss the main limitations of FOL approaches and motivate the need of a different semantic characterization based on epistemic logic, which is then precisely described in Section 6. In Section 7 we discuss the issues of modularity, generality and decidability under the two semantics. Finally, in Section 8 we highlight some open issues and challenging research directions.

2. State of the art

The main scientific base for P2P data integration is in traditional mediator-based data integration. The goal of mediator-based data integration systems is to provide clients with the access to data stored in heterogeneous and autonomous sources, without the need to know the physical characteristics of such sources and the precise location of the data.

As shown in Figure 1, a mediator-based data integration system exports to the user a global reconciled view of the data, called *global schema*, in terms of which the user formulates his/her queries, and the system maintains a declarative specification (i.e., a *mapping*) of the interrelationships between the global schema and the sources, often in turn represented through a *source schema* [67,58,55]. Two basic approaches for specifying the mapping have been proposed in the literature. The first approach, called *global-as-view* (GAV), requires that a view, i.e., a query, over the sources is associated with every element of the global schema, so that its meaning is specified in terms of the data residing at the sources. This is, for example the approach followed in [41,66,44,11]. Conversely, the second approach, called *local-as-view* (LAV), requires the sources to be defined as views over the global schema, i.e., it requires that a query over the global schema is associated to every source element. Examples of proposals following such an approach are [53,33,19]. More recently, a further approach has been considered, which

allows for specifying mapping assertions in which a query over the global schema is put in correspondence with a query over the source schema [55]. Such an approach, which is called *Global-local-as-view* (GLAV), since it generalizes both the LAV and the GAV approach [55], as so far received little attention in mediator-based data integration (whereas it has been recently investigated in P2P data integration, as we will see in the following sections).

Among the various problems related to data integration, the problem of answering queries posed over the global schema is the one that has been addressed most intensively. First proposals, developed in the middle 90s, faced such a problem in a procedural way, thus not providing the users with any declarative support to data integration. Systems like TSIMMIS (The Stanford-IBM Manager of Multiple Information Sources) [28], or Garlic [24] can be essentially considered as (simple) hierarchies of wrappers and mediators (and therefore can be both considered a primitive form of GAV systems). Wrappers are modules that hide the real nature of a data source, and present it and its data in a suitable format adopted within the system. Each wrapper manages the access to a single source and is in charge of translating queries over such a source in the specific language it uses, taking the answer the source returns, and providing them to the mediators. Each mediator is in charge of performing actual integration, by triggering the right wrappers, putting together the data that they return, and providing the final answer to users' requests (or feeding in turn other mediators). It has to be stressed that in TSIMMIS no global integration is ever performed, since each mediator works in an independent manner.

Systems like Information Manifold (IM) [59,60], or INFOMASTER [43,2,33] follow instead a more declarative approach. Such systems allow for the specification of a global schema, a source schema (both schemas are assumed to be relational), and a mapping between them, which for both systems is specified according to the LAV approach, whereas queries in the mapping are conjunctive queries, i.e., SQL select-project-join queries. For query processing, the Information Manifold system makes use of a procedure called the *bucket algorithm*, whereas the INFOMASTER system uses the *inverse rules algorithm*. Both algorithms solve query answering via query rewriting: a user query posed over the global schema is first suitably reformulated in a new query specified over the source schema, and then evaluated over the source extension in order to obtain the final answers. Several extensions have been proposed for both the algorithms. For example, in [33] the inverse rules algorithm is extended in order to handle users' queries specified in recursive Datalog, the presence of functional dependencies over the global schema, and the presence of limitations in accessing the sources (binding patterns), whereas in [64] an interesting optimization of the bucket algorithm has been proposed which significantly speeds up query processing.

A recent intensive investigation has been addressed the query answering problem for those cases in which integrity constraints (ICs) are specified over relational global schemas. ICs allow for enriching the representation of the integration domain, therefore constitute a powerful feature from a modeling point of view. However, they strongly affect the query answering process, since data stored at the sources may be in general incomplete or inconsistent with respect to such constraints. As for the first issue, query answering turns out to be a form of reasoning in the presence of incomplete information, suitably supported by a first-order interpretation of the system. This is the case of [49], which considers (limited combinations of) inclusion and functional dependencies in LAV data integration systems, or [10], where an algorithm for query answering in the presence

of key and foreign key constraints is provided, or [13] where a completely intensional procedure based on query rewriting is defined for all decidable cases in which key and inclusion dependencies are specified over the global schema. However, in those cases in which data may contradict global integrity constraints, the problem arises of how to obtain significative answers from inconsistent systems. Traditionally, the approach adopted to remedy to this problem has been through data cleaning [7]. This approach is procedural in nature, and is based on domain-specific transformation mechanisms applied to the data retrieved from the sources. Only very recently first academic prototype implementations have appeared, which provide declarative approaches to the treatment of inconsistency of data, in the line of the studies on consistent query answering [4]. In such approaches the common basic idea is that the inconsistency might be eliminated by modifying the database representing the extension of the system, and reasoning on the "repaired" database. Depending on the semantic assumption adopted for the system, several forms of repairing may be possible. Recently, several approaches to formalize repair semantics by using logic programs have been proposed [46,13,8]. The common idea is to encode the constraints of the global schema into a logic program, using unstratified negation or disjunction, such that the stable models of this program [42] yield the repairs of the global database. Among the most interesting proposals in data integration, we mention the INFOMIX system [56]. INFOMIX provides solutions for GAV data integration of heterogeneous data sources (e.g., relational, XML, HTML) accessed through relational global schemas over which powerful forms of integrity constraints can be issued (e.g., key, inclusion, and exclusion dependencies), and user queries are specified in a powerful query language (e.g., Datalog). The query answering technique proposed in such a system is based on query rewriting in Datalog enriched with negation and disjunction, under stable model semantics [13,48].

A setting similar to the one considered in INFOMIX is the one at the basis of the DIS@DIS system [14]. Even if limited in its capability of integrating sources with different data formats (the system actually considers only relational data sources), DIS@DIS however provides mechanisms also for integration of inconsistent data in LAV. In [9,8] an approach similar to the one followed in INFOMIX is followed, but a different repair semantics is adopted, which, to some extent, does not seem adequate to capture also incompleteness of data. Other interesting proposals on consistent query answering are the Hippo system [31,30], and the ConQuer system [38,40]. However, such proposals have been essentially developed in the context of a single database system, and therefore do not deal with all aspects of a complex data integration environment. Furthermore, w.r.t. classes of constraints and query language considered, the Hippo and the ConQuer systems are to some extent orthogonal to the INFOMIX and the DIS@DIS systems. They are geared towards highly efficient query answering for specific, polynomial-time classes of queries, whereas INFOMIX and DIS@DIS, instead, aim at supporting more general, highly expressive classes of queries (including also queries intractable under worst case complexity).

Many other studies have considered the query answering problem in data integration systems in various settings. For example, in [45,58] the relational setting under various assumptions on the languages used for the mapping and the queries has been analyzed, whereas in [18,19] query answering has been studied for the setting in which the global schema is formulated in an expressive conceptual data model. Also, query answering

in the presence of semistructured data sources and global schemas has been the subject of [21,23,22], and is still the subject of intensive investigations.

A different approach in mediator-based semantic interoperability looks at data management under the perspective of exchanging data between the sources and the global schema. Sources are again connected by means of mappings to the global schema, but in this case, the focus is on materializing the data flowing from the sources to the global schema. This problem is addressed in particular by the studies on Data Exchange. In short, Data Exchange is the problem of taking data structured under a source schema and creating an instance of a target schema (called solution) that reflects the source data as accurately as possible. Among several papers produced in the field, we mention [35,36,3], where data exchange is considered also in the presence of expressive constraints specified over the target schema, and powerful forms of mappings between the source and the target schema.

More recently, the issue of data integration, has been investigated in the more dynamic context of Peer-to-Peer (P2P) data computing [50]. In the last years, the P2P paradigm has been imposing in different contexts where the issue of cooperation, integration, and coordination between information nodes in a networked environment assumes a crucial role, including the Semantic Web [51], Grid computing, service oriented computing and distributed agent systems [63,52]. In all these systems, the problem of interoperability still needs deep investigation. In the following we review the main approaches proposed so far in the literature.

P2P systems have recently become popular for content sharing, and a number of different approaches have been studied to perform content retrieval in such networks (e.g., adaptation, deterministic placement of contents) [32,68]. In particular, the P2P paradigm was made popular by Napster, which employed a centralized database with references to the information items (files) on the peers. Gnutella, another well-known P2P system, has no central database, and is based on a communication-intensive search mechanism. More recently, a Gnutella-compatible P2P system, called Gridella [1], has been proposed, which follows the so-called Peer-Grid (P-Grid) approach. A P-Grid is a virtual binary tree that distributes replication over community of peers and supports efficient search. P-Grid's search structure is completely decentralized, supports local interactions between peers, uses randomized algorithms for access and search, and ensures robustness of search against node failures. As pointed out in [47], current P2P systems focus strictly on handling semantic-free, large-granularity requests for objects by identifier, which both limits their utility and restricts the techniques that might be employed to distribute the data. These current sharing systems are largely limited to applications in which objects are described by their name, and exhibit strong limitations in establishing complex links between peers. To overcome these limitations, data-oriented approaches to P2P have been proposed recently [5,50,6,47]. Some of them, see e.g., [69,5], are developed according to a super-peer based topology. A super-peer is a special node which manages a subset of client nodes. Such nodes interact only with the super-peer to which they are connected and receive results from it, whereas super-peers are also connected one another and communicate with other super-peers on behalf of their clients. In such systems, P2P computing is actually performed at the super-peer level, whereas communication between the super-peer and its clients is managed according to more traditional mediator-based techniques.

Conversely, other schema-based P2P systems do not require the presence of a super-peer. This is for example the case of the Piazza system [47], in which data origins serve original content, peer nodes cooperate to store materialized views and answer queries, nodes are connected by bandwidth-constrained links and advertise their materialized views to share resources with other peers. On the other hand, strong limitations on the topology of the mappings among peers are imposed by the system in order to allow for effective query answering.

However, apart from basic structure and algorithms, there is still a fundamental lack of understanding behind the basic issues of data integration in P2P systems, both from the point of view of modeling the system and characterizing its semantics, and from the point of view of computing answers to queries posed to a peer.

As for the modeling problem, it needs to be investigated whether the usual approach of resorting to a first-order logic interpretation of P2P mappings (followed, e.g., by [26, 50,6]), is still appropriate in the presence of an arbitrary structure of the system, possibly involving cycles among various nodes, or whether alternative semantic characterizations should be adopted [15]. As for the computational perspective, the basic task of computing query answers in P2P systems is still largely uninvestigated. Difficulties arise from the necessity of distributing the overall computation to the single nodes, exploiting their local processing capabilities and the underlying technological framework. Furthermore, query answering is in general related to the problem of finding a way to obtain answers relying only on the query answering services available at the peers. Each peer of the P2P system provides the service of answering queries expressed over its exported schema, and in general such services are the only basic services that we can rely upon in order to answer queries.

The problem is even more complex when peers export an ontology (rather than a simple relational schema) [65,16]. Here, the problem of how to exploit the mappings between peers in order to answer queries posed to one peer is in general hard to solve, even in very simple settings (e.g., when the whole system is constituted by two interoperating peers as in [16]). Indeed, query answering in this setting peers is actually a complex form of query reformulation. Notice that this problem is crucial in several contexts, as, for example mediator-based data integration, in particular in the case where the global schema is expressed as an ontology. Also, recent studies on query rewriting under integrity constraints, some of that we discussed before [13,10], are strictly related to such a form of query rewriting. Then, this problem is of clear relevance for the Semantic Web, even if research on the Semantic Web has focused more on the problem of ontology matching (i.e., finding the mapping between peers).

Analogously to the case of mediator-based data integration, in the P2P architecture a different approach to achieve cooperation between different peers can be the one of exchanging data between peers. Peers are again interconnected by means of mappings, but in this case, the focus is on materializing the data flowing from one peer to another. Whereas traditional Data Exchange has been the subject of several recent investigations, P2P Data Exchange has so far received little attention. In [39] the problem of deciding the existence of a solution and establishing computational complexity of such a decision process is addressed in Peer Data Exchange, a setting in which only two peers interact that have different roles and capabilities. However, Data Exchange in a full-fledged P2P setting remains still unexplored.

3. Framework

In our work, we use the framework for P2P data integration presented in [20], which is briefly described in this section.

We refer to a fixed, infinite, denumerable set Γ of constants. Such constants are shared by all peers, and denote the data items managed by the Peer-to-Peer Data Integration System (P2PDIS). Moreover, given a relational alphabet A, we denote with $\mathcal{L}_A$ the set of function-free first-order logic formulas whose relation symbols are in A and whose constants are in Γ.

We also consider conjunctive queries, i.e., SQL select-project-join queries. Formally, a *conjunctive query* (CQ) of arity n over A is a query 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* $\mathcal{P} = \{P_1, \ldots, P_n\}$ is constituted by a set of n peers. Each peer $P_i \in \mathcal{P}$ (cf. [50]) is defined as a tuple $P_i = (id, G, S, L, M, \mathcal{L})$, where:

- id is a symbol that identifies the peer P_i within $\mathcal{P}$, called the identifier of P_i.
- G is the *schema* of P_i, which is a finite set of formulas of $\mathcal{L}_{A_G}$ (representing local integrity constraints), where A_G is a relational alphabet (disjoint from the other alphabets in $\mathcal{P}$) called the *alphabet* of P_i. Intuitively, the peer schema provides an intensional view of the information managed by the peer.
- S is the *(local) source schema* of P_i, which is simply a finite relational alphabet (again disjoint from the other alphabets in $\mathcal{P}$), called the *local alphabet* of P_i. Intuitively, the source schema describes the structure of the data sources of the peer (possibly obtained by wrapping physical sources), i.e., the sources where the real data managed by the peer are stored.
- 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, respectively over the source schema S and over the peer schema G. The local mapping assertions establish the connection between the elements of the source schema and those of the peer schema in P_i. 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. In the terminology used in data integration, the combination of peer schema, source schema, and local mapping assertions constitutes a GLAV *data integration system* [55] 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*, which specify the semantic relationships that the peer P_i has with the other peers. Each assertion in M is an expression of the form

  $$cq' \rightsquigarrow cq,$$

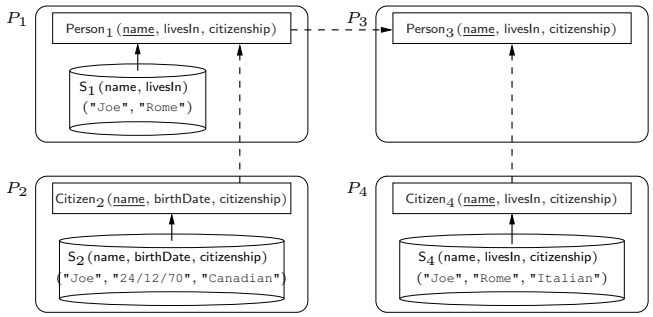

Figure 3. A P2P system

where cq, called the *head* of the assertion, is a conjunctive query over the peer (schema of) P_i, while 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 $\mathcal{P}$. A P2P mapping assertion $cq' \rightsquigarrow cq$ from peer P_j to peer P_i expresses the fact that the P_j-concept represented by cq' is mapped to the P_i-concept represented by cq. From an extensional point of view, the assertion specifies that every tuple that can be retrieved from P_j by issuing query cq' satisfies cq in P_i. Observe that no limitation is imposed on the topology of the whole set of P2P mapping assertions in the system $\mathcal{P}$, and hence, as in [20], the set of all P2P mappings may be cyclic.

- $\mathcal{L}$ is a relational query language specifying the class of queries that the peer P_i can process. We assume that $\mathcal{L}$ is some fragment of FOL that accepts at least conjunctive queries. We say that the queries in $\mathcal{L}$ are those *accepted by* P_i. Notice that this implies that, for each P2P mapping assertion $cq' \rightsquigarrow cq$ from another peer P_j to peer P_i in M, we have that cq' is accepted by P_j.

An *extension* for a P2PDIS $\mathcal{P} = \{P_1, \ldots, P_n\}$ is a set $\mathcal{D} = \{D_1, \ldots, D_n\}$, where each D_i is an extension of the predicates in the local source schema of peer P_i.

A P2PDIS, together with an extension, is intended to be queried by external users. A user enquires the whole system by accessing any peer P of $\mathcal{P}$, and by issuing a *query* q to P. The query q is processed by P if and only if q is expressed over the schema of P and is accepted by P.

Example 3.1 Let us consider the P2PDIS in Figure 3, in which we have 4 peers P_1, P_2, P_3, and P_4 (in the following, we assume that each peer P_i is identified by i).

The global schema of peer P_1 is formed by a relation schema $\mathsf{Person}_1(\underline{\mathsf{name}}, \mathsf{livesIn}, \mathsf{citizenship})$, where name is the key (we underline the key of a relation). P_1 contains a local source $\mathsf{S}_1(\mathsf{name}, \mathsf{livesIn})$, mapped to the global view by the assertion $\{x, y \mid \mathsf{S}_1(x, y)\} \rightsquigarrow \{x, y \mid \exists z.\, \mathsf{Person}_1(x, y, z)\}$. Moreover, it has a P2P mapping assertion $\{x, z \mid \exists y.\, \mathsf{Citizen}_2(x, y, z)\} \rightsquigarrow \{x, z \mid \exists y.\, \mathsf{Person}_1(x, y, z)\}$ relating information in peer P_2 to those in peer P_1.

P_2 has $\mathsf{Citizen}_2(\underline{\mathsf{name}}, \mathsf{birthDate}, \mathsf{citizenship})$ as global schema, and a local source $\mathsf{S}_2(\mathsf{name}, \mathsf{birthDate}, \mathsf{citizenship})$ mapped to the global schema through the local mapping $\{x, y, z \mid \mathsf{S}_2(x, y, z)\} \rightsquigarrow \{x, y, z \mid \mathsf{Citizen}_2(x, y, z)\}$. P_2 has no P2P mappings.

P_3 has $\mathsf{Person}_3(\underline{\mathsf{name}}, \mathsf{livesIn}, \mathsf{citizenship})$ as global schema, contains no local sources, and has a P2P mapping $\{x, y, z \mid \mathsf{Person}_1(x, y, z)\} \rightsquigarrow \{x, y, z \mid$

$\mathsf{Person}_3(x, y, z)\}$ with P_1, and a P2P mapping $\{x, y, z \mid \mathsf{Citizen}_4(x, y, z)\} \rightsquigarrow \{x, y, z \mid \mathsf{Person}_3(x, y, z)\}$ with P_4.

P_4 has $\mathsf{Citizen}_4(\underline{\mathsf{name}}, \mathsf{livesIn}, \mathsf{citizenship})$ as global schema, and a local source $\mathsf{S}_4(\mathsf{name}, \mathsf{livesIn}, \mathsf{citizenship})$ mapped to the global schema through the local mapping $\{x, y, z \mid \mathsf{S}_4(x, y, z)\} \rightsquigarrow \{x, y, z \mid \mathsf{Citizen}_4(x, y, z)\}$. P_4 has no P2P mappings.

Finally, Figure 1 shows also an extension of the P2P data integration system, which includes $\mathsf{S}_1(\texttt{"Joe"}, \texttt{"Rome"})$, $\mathsf{S}_2(\texttt{"Joe"}, \texttt{"24/12/70"}, \texttt{"Canadian"})$, and $\mathsf{S}_4(\texttt{"Joe"}, \texttt{"Rome"}, \texttt{"Italian"})$. ∎

4. Classical semantics for P2P data integration systems

In this section we present a logical formalization of P2P data integration systems based on classical first-order logic. Such a formalization is the first one that has been proposed for P2P data integration [26,54,50].

We assume that the peers are interpreted over a fixed infinite domain Δ. We also fix the interpretation of the constants in Γ (cf. previous section) so that: (i) each $c \in \Gamma$ denotes an element $d \in \Delta$; (ii) different constants in Γ denote different elements of Δ; (iii) each element in Δ is denoted by a constant in Γ.[1] 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.

4.1. Semantics of one peer

We focus first on the semantics of a single peer $P = (id, G, S, L, M, \mathcal{L})$. Let us call *peer theory of* P the FOL theory T_P defined as follows. The alphabet of T_P is obtained as union of the alphabet 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

$$\forall \mathbf{x}. (\exists \mathbf{y}. \mathit{body}_{cq_S}(\mathbf{x}, \mathbf{y}) \supset \exists \mathbf{z}. \mathit{body}_{cq_G}(\mathbf{x}, \mathbf{z}))$$

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 we only have 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 it is a model of the FOL theory T_P such that 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 $\mathcal{L}$ 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 $\mathcal{L}$. 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 based on D, we have that $\mathbf{t} \in q^{\mathcal{I}}$.

[1] In other words the constants in Γ act as *standard names* [57].

4.2. Semantics for P2P data integration systems

Based on the above logical formalization of a peer, we now present the "classical" approach to providing a semantics to the whole P2P data integration system. The classical approach is what we may call the FOL approach, followed by [26,54,50]. 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 (P2P mappings are not considered in $T_{\mathcal{P}}$).

By following 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 for each relational symbol s of the source schemas in the peers of $\mathcal{P}$, we have that $s^{\mathcal{I}} = s^{\mathcal{D}}$. Then we call *FOL model of $\mathcal{P}$ based on $\mathcal{D}$* a model $\mathcal{I}$ of $T_{\mathcal{P}}$ based on $\mathcal{D}$ that 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 of the peers P in $\mathcal{P}$ (assuming that the identifier of P is id) and a source database $\mathcal{D}$ for $\mathcal{P}$, we define the *certain answers* $ANS_{fol}(q, id, \mathcal{P}, \mathcal{D})$ 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 $\mathbf{t} \in q^{\mathcal{I}}$.

5. Limitations of first-order approaches

Although correct from a formal point of view, the usual approach of resorting to a first-order logic interpretation of P2P mappings, which we have described in the above section, has several drawbacks, both from the modeling and from the computational perspective. Consider, for example, three central desirable properties of P2P systems:

- *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 (P2P 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.

Actually, these desirable properties are weakly supported by approaches based directly on FOL semantics. Indeed, such approaches essentially consider the P2P system as a single flat logical theory. As a result, the structure of the system in terms of peers is lost and remote interconnections may propagate constraints that have a deep impact on the semantics of a peer. Moreover, under arbitrary P2P interconnections, query answering under the first-order semantics is undecidable, even when the single peers have

an extremely restricted structure. Motivated by these observations, several authors proposed suitable limitations to the form of P2P mappings, such as acyclicity, thus giving up generality to retain decidability [50,54,35].

To overcome the above drawbacks, we propose a new semantics for P2P systems, with the following aims:

- We want to take into account that peers in our context 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.
- We seek for a semantic characterization that leads to a setting where query answering is decidable, and possibly, polynomially tractable.

We base our proposal of a new semantics for P2P systems on epistemic logic, and we show that the new semantics is clearly superior to the usual FOL semantics with respect to all three properties mentioned above.

6. Multi-modal epistemic formalization

In this section we present a logical formalization of P2P data integration systems. Although one possible choice for formalizing such systems is classical first order logic, it was argued in [20] that using epistemic logic brings several advantages. In particular, we adopt a *multi-modal* epistemic logic, based on the premise that each peer in the system can be seen as a rational agent. More precisely, the formalization we provide in this section is based on $\mathsf{S5}_n$, the multi-modal version of the modal logic S5 [29,57].

6.1. The logic $\mathsf{S5}_n$

The language $\mathcal{L}(\mathsf{S5}_n)$ of $\mathsf{S5}_n$ is obtained from first-order logic by adding a set $\mathbf{K_1}, \ldots, \mathbf{K_n}$ of modal operators, for the forming rule: if ϕ is a (possibly open) formula, then also $\mathbf{K_i}\phi$ is so, for $1 \leq i \leq n$ for a fixed n. In $\mathsf{S5}_n$, each modal operator is used to formalize the epistemic state of a different agent. Informally, the formula $\mathbf{K_i}\phi$ should be read as "ϕ is known to hold by the agent i". The semantics of $\mathsf{S5}_n$ is such that what is known by an agent must hold in the real world: in other words, the agent cannot have inaccurate knowledge of what is true, i.e., believe something to be true although in reality it is false. Moreover, $\mathsf{S5}_n$ states that the agent has complete information on what it knows, i.e., if agent i knows ϕ then it knows of knowing ϕ, and if agent i does not know ϕ, then it knows that it does not know ϕ. In other words, the following assertions hold for every $\mathsf{S5}_n$ formula ϕ:

$\mathbf{K_i}\phi \supset \phi$	known as the axiom schema T
$\mathbf{K_i}\phi \supset \mathbf{K_i}(\mathbf{K_i}\phi)$	known as the axiom schema 4
$\neg\mathbf{K_i}\phi \supset \mathbf{K_i}(\neg\mathbf{K_i}\phi)$	known as the axiom schema 5

To define the semantics of $\mathsf{S5}_n$, we start from first-order interpretations. In particular, we restrict our attention to first-order interpretations that share a fixed infinite domain

Δ. We further assume that for each domain element $d \in \Delta$, we have a unique constant $c_d \in \Gamma$ that denotes exactly d, and, vice versa, that every constant $c_d \in \Gamma$ denotes exactly one domain element $d \in \Delta$[2].

Formulas of $\mathsf{S5}_n$ are interpreted over $\mathsf{S5}_n$-structures. A $\mathsf{S5}_n$-*structure* is a Kripke structure E of the form $(W, \{R_1, \ldots R_n\}, V)$, where: W is a set whose elements are called *possible worlds*; V is a function assigning to each $w \in W$ a first-order interpretation $V(w)$; and each R_i, called the *accessibility relation* for the modality $\mathbf{K_i}$, is a binary relation over W, with the following constraints:

if $w \in W$ then $(w, w) \in R_i$, i.e., R_i is reflexive
if $(w_1, w_2) \in R_i$ and $(w_2, w_3) \in R_i$ then $(w_1, w_3) \in R_i$, i.e., R_i is transitive
if $(w_1, w_2) \in R_i$ and $(w_1, w_3) \in R_i$ then $(w_2, w_3) \in R_i$, i.e., R_i is euclidean

An $\mathsf{S5}_n$-*interpretation* is a pair E, w, where $E = (W, \{R_1, \ldots R_n\}, V)$ is an $\mathsf{S5}_n$-structure, and w is a world in W. We inductively define when a sentence (i.e., a closed formula) ϕ *is true in an interpretation* E, w (or, is true on world $w \in W$ in E), written $E, w \models \phi$, as follows:[3]

$E, w \models P(c_1, \ldots, c_n)$	iff	$V(w) \models P(c_1, \ldots, c_n)$
$E, w \models \phi_1 \wedge \phi_2$	iff	$E, w \models \phi_1$ and $E, w \models \phi_2$
$E, w \models \neg\phi$	iff	$E, w \not\models \phi$
$E, w \models \exists x.\, \psi$	iff	$E, w \models \psi_c^x$ for some constant c
$E, w \models \mathbf{K_i}\phi$	iff	$E, w' \models \phi$ for every w' such that $(w, w') \in R_i$

We say that a sentence ϕ is *satisfiable* if there exists an $\mathsf{S5}_n$-*model* for ϕ, i.e., an $\mathsf{S5}_n$-interpretation E, w such that $E, w \models \phi$, *unsatisfiable* otherwise. A *model* for a set Σ of sentences is a model for every sentence in Σ. A sentence ϕ is *logically implied* by a set Σ of sentences, written $\Sigma \models_{\mathsf{S5}_n} \phi$, if and only if in every $\mathsf{S5}_n$-model E, w of Σ, we have that $E, w \models \phi$.

Notice that, since each accessibility relation of a $\mathsf{S5}_n$-structure is reflexive, transitive and Euclidean, all instances of axiom schemas T, 4 and 5 are satisfied in every $\mathsf{S5}_n$-interpretation.

6.2. Epistemic semantics for P2P data integration systems

Due to the characteristics mentioned above, $\mathsf{S5}_n$ is well-suited to formalize P2PDISs of the kind presented in Section 3. Let $\mathcal{P} = \{P_1, \ldots, P_n\}$ be a P2PDIS in which each peer P_i has identifier i. For each peer $P_i = (i, G, S, L, M, \mathcal{L})$ we define the theory $\mathcal{T}_K(P_i)$ in $\mathsf{S5}_n$ as the union of the following sentences:

- Global schema G of P_i: for each sentence ϕ in G, we have

  $$\mathbf{K_i}\phi$$

  Observe that ϕ is a first-order sentence expressed in the alphabet of P_i, which is disjoint from the alphabets of all the other peers in $\mathcal{P}$.

[2] In other words, the constants in Γ act as *standard names* [57].

[3] We have used ψ_c^x to denote the formula obtained from ψ by substituting each free occurrence of the variable x with the constant c.

- Local mapping assertions L between G and the local source schema S: for each mapping assertion $\{\mathbf{x} \mid \exists \mathbf{y}.\, body_{cq_S}(\mathbf{x}, \mathbf{y})\} \rightsquigarrow \{\mathbf{x} \mid \exists \mathbf{z}.\, body_{cq_G}(\mathbf{x}, \mathbf{z})\}$ in L, we have

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

- P2P mapping assertions M: for each P2P mapping assertion $\{\mathbf{x} \mid \exists \mathbf{y}.\, body_{cq_j}(\mathbf{x}, \mathbf{y})\} \rightsquigarrow \{\mathbf{x} \mid \exists \mathbf{z}.\, body_{cq_i}(\mathbf{x}, \mathbf{z})\}$ between the peer j and the peer i in M, we have

$$\forall \mathbf{x}.\, \mathbf{K_j}(\exists \mathbf{y}.\, body_{cq_j}(\mathbf{x}, \mathbf{y})) \supset \mathbf{K_i}(\exists \mathbf{z}.\, body_{cq_i}(\mathbf{x}, \mathbf{z})) \quad (1)$$

  In words, this sentence specifies the following rule: for each tuple of values $\mathbf{t}$, if peer j knows the sentence $\exists \mathbf{y}.\, body_{cq_j}(\mathbf{t}, \mathbf{y})$, then peer i knows the sentence $\exists \mathbf{z}.\, body_{cq_i}(\mathbf{t}, \mathbf{z})$ holds.

We denote by $\mathcal{T}_K(\mathcal{P})$ the theory corresponding to the P2PDIS $\mathcal{P}$, i.e., $\mathcal{T}_K(\mathcal{P}) = \bigcup_{i=1,\ldots,n} \mathcal{T}_K(P_i)$.

Example 6.1 We provide now the formalization of the P2PDIS of Example 3.1. The theory $\mathcal{T}_K(P_1)$ modeling peer P_1 is the conjunction of:

$$\mathbf{K_1}(\forall x, y, y', z, z'.\, \mathsf{Person}_1(x, y, z) \wedge \mathsf{Person}_1(x, y', z') \supset y = y' \wedge z = z')$$
$$\mathbf{K_1}(\forall x, y.\, \mathsf{S}_1(x, y) \supset \exists z.\, \mathsf{Person}_1(x, y, z))$$
$$\forall x, z.\, \mathbf{K_2}(\exists y.\, \mathsf{Citizen}_2(x, y, z)) \supset \mathbf{K_1}(\exists y.\, \mathsf{Person}_1(x, y, z))$$

The theory $\mathcal{T}_K(P_2)$ modeling peer P_2 is the conjunction of:

$$\mathbf{K_2}(\forall x, y, y', z, z'.\, \mathsf{Citizen}_2(x, y, z) \wedge \mathsf{Citizen}_2(x, y', z') \supset y = y' \wedge z = z')$$
$$\mathbf{K_2}(\forall x, y, z.\, \mathsf{S}_2(x, y, z) \supset \mathsf{Citizen}_2(x, y, z))$$

The theory $\mathcal{T}_K(P_3)$ modeling peer P_3 is the conjunction of:

$$\mathbf{K_3}(\forall x, y, y', z, z'.\, \mathsf{Person}_3(x, y, z) \wedge \mathsf{Person}_3(x, y', z') \supset y = y' \wedge z = z')$$
$$\forall x, y.\, \mathbf{K_1}(\exists z.\, \mathsf{Person}_1(x, z, y)) \supset \mathbf{K_3} \exists z.\, \mathsf{Person}_3(x, z, y)$$
$$\forall x, y, z.\, \mathbf{K_4}(\mathsf{Citizen}_4(x, y, z)) \supset \mathbf{K_3} \mathsf{Person}_3(x, y, z)$$

The theory $\mathcal{T}_K(P_4)$ modeling peer P_4 is the conjunction of:

$$\mathbf{K_4}(\forall x, y, y', z, z'.\, \mathsf{Citizen}_4(x, y, z) \wedge \mathsf{Citizen}_4(x, y', z') \supset y = y' \wedge z = z')$$
$$\mathbf{K_4}(\forall x, y, z.\, \mathsf{S}_4(x, y, z) \supset \mathsf{Citizen}_4(x, y, z))$$

■

The extension $\mathcal{D} = \{D_1, \ldots, D_n\}$ of a P2PDIS $\mathcal{P}$ is modeled as a sentence constituted by the conjunction of all facts corresponding to the tuples stored in the sources, i.e., $DB(\mathcal{D}) = \bigwedge_{i=1}^{n} DB(D_i)$ where $DB(D_i) = \mathbf{K_i}(\bigwedge_{t \in r^{D_i}} r(t))$.

A client of the P2PDIS interacts with one of the peers, say peer P_i, posing a *query* to it. A query q is an open formula $q(\mathbf{x})$ with free variables $\mathbf{x}$ expressed in the language accepted by the peer P_i (we recall that such a language is a subset of first-order logic). The semantics of a query $q \in \mathcal{L}$ posed to a peer $P_i = (i, G, S, L, M, \mathcal{L})$ of $\mathcal{P}$ with respect to an extension $\mathcal{D}$ is defined as the set of tuples

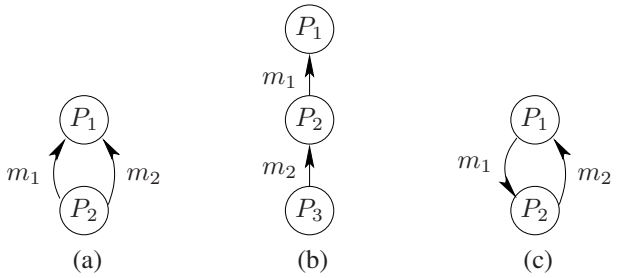

Figure 4. Interactions between two mappings

$$ANS_{\mathsf{S5}_n}(q, i, \mathcal{P}, \mathcal{D}) = \{\mathbf{t} \mid \mathcal{T}_K(\mathcal{P}) \cup DB(\mathcal{D}) \models_{\mathsf{S5}_n} \mathbf{K_i} q(\mathbf{t})\}$$

where $q(\mathbf{t})$ denotes the sentence obtained from the open formula $q(\mathbf{x})$ by replacing all occurrences of the free variables in $\mathbf{x}$ with the corresponding constants in $\mathbf{t}$.

6.3. Query answering in P2PDISs under the epistemic semantics

Finally, we point out that query answering in P2P data integration systems under the epistemic semantics described above has been studied in [15,20]. In particular, it has been proved that, under very general assumptions about the structure of the single peers and the query language, query answering in a P2PDIS under the epistemic semantics is decidable and, in many cases, tractable in data complexity, i.e., queries can be answered in polynomial time with respect to the size of the data stored at the peers.

In the above epistemic setting, two different query answering strategies have been explored. In the first one, described in [15], a bottom-up algorithm has been defined, which is able to answer a query by incrementally collecting the certain answers to suitable queries that are iteratively computed at the various peers, following the topology of the mappings among the peers. The second approach, presented in [20], adopts a top-down technique that is heavily based on query reformulation: in particular, the peer to which the initial query is posed incrementally collects the global reformulation of the query, which is computed by a distributed algorithm that exploits both the mappings among the peers and a local query reformulation ability of each peer. We refer the interested reader to [15,20] for more details.

7. Comparison between the two semantics

In this section, we compare the epistemic and FOL semantics for P2P systems presented above. The comparison is guided by three principles, namely modularity, generality, and decidability of query answering. To highlight the differences between the two semantics, we will consider the simplest setting in which interactions may occur, namely systems containing only two P2P mappings. The three types of systems we discuss in the following are depicted in Figure 4, 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. For the sake of simplicity, in the following we slightly simplify the formalization, and denote a peer P_i by a quadruple $P_i = (G, S, L, M)$, i.e., with respect to the notation used above, we omit the first component id (the identifier of P), and the last component $\mathcal{L}$ (the class of queries that the peer P can process), and assume that: (i) the identifier of peer P_i is the subscript i of the peer symbol; (ii) all peers accept the same query language $\mathcal{L}$, and such an $\mathcal{L}$ is the language of conjunctive queries.

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 system $\mathcal{P}$, we denote as $\tau(\mathcal{P})$ the P2P 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.

7.1. Modularity: Parallel composition

We consider a P2P system $\mathcal{P}_{par}$ with the structure depicted in Figure 4(a), and to highlight the interdependence between mappings, we further 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 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 $ANS_X(q, 1, \mathcal{P}, \{D_2\})$ coincides with the certain answers $ANS(q, \tau(P_1), \delta(P_1, D_2))$ over $\tau(P_1)$. The following theorems show that a P2P system as simple as $\mathcal{P}_{par}$ is sufficient to separate the epistemic and the FOL semantics with respect to modularity.

Theorem 7.1 *There is a P2P 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 $ANS_{fol}(q, 1, \mathcal{P}, \{D_2\}) \neq ANS(q, \tau(P_1), \delta(P_1, D_2))$.*

Proof (sketch). We exhibit $\mathcal{P}_{par} = \{P_1, P_2\}$, D_2, and q such that the claim holds. Let $P_1 = (\{u/1\}, \emptyset, \emptyset, \{m_1, m_2\})$ and $P_2 = (G_2, \{s/1\}, \{\ell_2\}, \emptyset)$, with $G_2 = \{\forall x\,(u_3(x) \supset u_1(x) \vee u_2(x))\}$, and

$$\begin{aligned} \ell_2 &= \{x \mid s(x)\} \rightsquigarrow \{x \mid u_3(x)\} \\ m_1 &= \{x \mid u_1(x)\} \rightsquigarrow \{x \mid u(x)\} \\ m_2 &= \{x \mid u_2(x)\} \rightsquigarrow \{x \mid u(x)\} \end{aligned}$$

Consider the source database $D_2 = \{s(a)\}$ for P_2. It is easy to see that for the query $q = \{x \mid u(x)\}$ we have that $ANS_{fol}(q, 1, \mathcal{P}, \{D_2\}) = \{a\}$, while $\delta(P_1, D_2) = \emptyset$, and hence $ANS(q, \tau(P_1), \delta(P_1, D_2)) = \emptyset$. □

For the epistemic semantics, from the results in [20], we get the following theorem.

Theorem 7.2 *Let $\mathcal{P}_{par}$ and D_2 be as above. Then, for every query q over P_1 we have that $ANS_{\mathsf{S5}_n}(q, 1, \mathcal{P}, \{D_2\}) = ANS(q, \tau(P_1), \delta(P_1, D_2))$.*

7.2. Modularity: Sequential composition

We consider a P2P system $\mathcal{P}_{seq}$ with the structure depicted in Figure 4(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 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 $ANS_X(q, 1, \mathcal{P}, \{D_3\}) = ANS(q, \tau(P_1), \delta(P_1, P_2, D_3))$.

We show that also in the context of sequential composition, while the epistemic semantics for P2P systems is modular, the FOL semantics is not so.

Theorem 7.3 *There is a P2P 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 $ANS_{fol}(q, 1, \mathcal{P}, \{D_3\}) \neq ANS(q, \tau(P_1), \delta(P_1, P_2, D_3))$.*

Proof (sketch). Exploiting a result in [61], we exhibit $\mathcal{P}_{seq} = \{P_1, P_2, P_3\}$, D_3, and q such that the claim holds. Let $P_1 = (\{u/2\}, \emptyset, \emptyset, \{m_1\})$, $P_2 = (\{v/2\}, \emptyset, \emptyset, \{m_2\})$, and $P_3 = (\{w/2\}, \{s/2\}, \{\ell_2\}, \emptyset)$, with

$$\begin{aligned} m_1 &= \{x, y \mid \exists z_1, z_2\,(v(x, z_1) \wedge v(z_1, z_2) \wedge v(z_2, y))\} \rightsquigarrow \{x, y \mid u(x, y)\} \\ m_2 &= \{x, y \mid w(x, y)\} \rightsquigarrow \{x, y \mid \exists z\,(v(x, z) \wedge v(z, y))\} \\ \ell_2 &= \{x, y \mid s(x, y)\} \rightsquigarrow \{x, y \mid w(x, y)\} \end{aligned}$$

Consider the source database $D_3 = \{s(a_i, a_{i+1}) \mid 1 \leq i \leq 7\}$ for P_2. It is easy to see that for the query $q = \{x, y \mid \exists z\,(u(x, z) \wedge u(z, y))\}$ we have that $ANS_{fol}(q, 1, \mathcal{P}, \{D_3\}) = \{(a_1, a_7)\}$, while $ANS(q, \tau(P_1), \delta(P_1, P_2, D_3)) = \emptyset$. □

For the epistemic semantics, from the results in [20], we get the following theorem.

Theorem 7.4 *Let $\mathcal{P}_{seq}$ and D_3 be as above. Then, for every query q over P_1 we have that $ANS_{\mathsf{S5}_n}(q, 1, \mathcal{P}, \{D_3\}) = ANS(q, \tau(P_1), \delta(P_1, P_2, D_3))$.*

7.3. Decidability: Cycle between two peers

We consider a P2P system $\mathcal{P}_{cyc}$ with the structure depicted in Figure 4(c). The presence of a cycle between two peers suffices to make query answering undecidable under the FOL semantics.

Theorem 7.5 *There is a P2P system $\mathcal{P}_{cyc} = \{P_1, P_2\}$ of the form as above, a source database $\mathcal{D}$ for $\mathcal{P}_{cyc}$, such that computing the certain answers to queries over the single peers P_1 and P_2 is decidable, while computing the certain answers to queries in $\mathcal{P}_{cyc}$ based on $\mathcal{D}$ under the FOL semantics is undecidable.*

Proof (sketch). The undecidability result follows by a reduction from undecidability of query answering under inclusion and functional dependencies [62,27,12].

Consider a relational schema $\mathcal{R}$ with inclusion and functional dependencies. We construct the peer $P_1 = (G_1, S_1, L_1, M_1)$ as follows: G_1 contains the relations of $\mathcal{R}$, plus two additional relations inc and fun, both containing one attribute $r.A$ for each attribute A in a relation r of $\mathcal{R}$. G_1 contains all inclusion assertion of $\mathcal{R}$, plus one inclusion assertion $r[\mathbf{A}, \mathbf{B}] \subseteq inc[r.\mathbf{A}, r.\mathbf{B}]$ and one functional dependency $fun : r.\mathbf{A} \rightarrow r.\mathbf{B}$, for each functional dependency $r : \mathbf{A} \rightarrow \mathbf{B}$ in $\mathcal{R}$ (we have denoted by $r.\mathbf{A}$ the tuple of attributes corresponding to $\mathbf{A}$). S_1 contains a source relation s_r for each relation r in $\mathcal{R}$, and L_1 maps such relations to the corresponding relations in G_1. M_1 contains a single P2P mapping assertion $\{\mathbf{x} \mid inc(\mathbf{x})\} \rightsquigarrow \{\mathbf{x} \mid rem(\mathbf{x})\}$.

Then we construct the peer $P_2 = (G_2, \emptyset, \emptyset, M_2)$, where G_2 contains only the relation rem (of the same arity as inc and fun), and M_2 contains a single P2P mapping assertion $\{\mathbf{x} \mid rem(\mathbf{x})\} \rightsquigarrow \{\mathbf{x} \mid fun(\mathbf{x})\}$.

Notice that query answering in P_1 is decidable, since all functional dependencies are on the relation fun, which is not related through inclusion dependencies to the other relations in G_1, and the implication problems and query answering problems for inclusion and functional dependencies separately are decidable [25,12]. Also, P_2 is trivially decidable. On the other hand, under the FOL semantics, the P2P mappings propagate the functional dependencies on fun to inc, and hence in turn to the relations in G_1. Therefore, the whole set of dependencies in $\mathcal{R}$ are reflected in G_1, thus making query answering in the P2P system as a whole undecidable. □

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 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 [50,54,35]. In practice this may even be unfeasible, again since no actor is in control of all P2P mappings.

On the other hand, under the epistemic semantics we can retain both generality and decidability for P2P systems with *arbitrary* structure, as shown in [15,20].

Theorem 7.6 *For each P2P system $\mathcal{P}$ of the form as above and a source database $\mathcal{D}$ for $\mathcal{P}$ such that computing the certain answers to queries over the single peers is decidable, computing the certain answers to queries in $\mathcal{P}$ based on $\mathcal{D}$ under the epistemic semantics is decidable.*

8. Future research directions and conclusions

In this chapter we have provided a formal framework for P2P data integration, and we have proposed a new semantics for such a framework based on epistemic logic. We have compared such a semantics with the commonly adopted semantics based on first-order logic, and we have shown that the epistemic approach is superior to the first-order one with respect to three central properties for P2P systems, namely, modularity, generality, and decidability. We have also devised a polynomial time query answering algorithm that is sound and complete for the problem of computing certain answers to user queries accepted by peers in the system (i.e., computing the set ANS_{S5_n}). The in details of the algorithm are described in [20].

We point out that our formalization and our query answering algorithm are among the first results on advanced P2P data integration, and that research and technology on such field is still at an embryonic state. Actually, a number of issues need to be investigated and resolved in order to facilitate powerful, human level P2P information integration. Among them, we consider the following ones the most challenging and important:

- Inconsistency tolerance, i.e., developing suitable mechanisms for the management of data inconsistency. Inconsistency in P2P data integration systems may arise for different reasons: a peer may be locally inconsistent because its data (possibly coming from local sources) violate integrity constraints specified on its schema; data coming into a peer from other peers may contradict constraints when combined with data locally managed by the peer; data coming into a peer from different peers may result mutually inconsistent, i.e., combined together may violate integrity constraints of the peer schema. Hence, the main issue is to deal appropriately with inconsistencies, in order to avoid trivialized query results. First results on this issue recently appeared in [17].
- Preferences and trust management, i.e., choosing from data coming from different peers on the basis of preferences/trust specifications. In P2P data integration systems, each peer may want to specify information on how it trusts peers, with respect to certain kind of data, to which it is connected, e.g., by assigning different quality values (e.g., reliability, availability, etc.) to data coming from different peers. Obviously, the "best" data should be preferred to the others. Preference/trust specifications can be used both to solve data inconsistencies and to rank data belonging to query answers. Indeed, when dealing with mutually inconsis-

tent data coming from different peers, a peer may choose to trust only those data coming from the preferred peer (e.g., the most reliable one). Furthermore, a peer may choose to select which data have to be first returned in the answer to a query, on the basis of preference/trust assertions, in order to speed up query processing. Also, preference specifications can be used during the query answering process to avoid or delay access to peers that have a low preference degree.

- Authorization and privacy management, i.e., controlling the accessibility of the data in the peers. In general, each peer in the system wants to allow only controlled accesses to its data, i.e., the peer needs the ability of specifying privacy or confidentiality for (a portion of) its data, or guaranteeing disclosure of certain information from particular users. An interesting approach which follows a logical perspective has been recently proposed in [70] in the context of a single database. The proposed method is based on the idea of specifying, for each user, a set of authorized views, representing the information that the user is allowed to access. We think that this idea nicely captures the logical essence of access control, and might be somehow transferred in the context of P2P data integration, where it is crucial that each peer is able to specify data privacy policy in a declarative way.
- Model management, i.e., methods for dealing with the dynamics of a P2P data integration system, in which peers may join and leave at any time, or be temporarily unavailable. This requests for both advanced meta-data management and handling partial and incomplete information.
- Data exchange, i.e., dealing with the materialization of data flowing from one peer to another. Materialized data can be profitably used in P2P data management in several ways, e.g., can be exploited to reduce the number of accesses to remote peers during query processing. Notice that whereas traditional data exchange has been the subject of several recent investigations (see Section 2 for a brief description), P2P data exchange has so far received little attention, and it still remains largely unexplored.

All these issues are unresolved to date and require major efforts and advances in future research in the field.

References

[1] K. Aberer, M. Punceva, M. Hauswirth, and R. Schmidt. Improving data access in P2P systems. *IEEE Internet Computing*, 2002.

[2] S. Abiteboul and O. Duschka. Complexity of answering queries using materialized views. In *Proc. of the 17th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS'98)*, pages 254–265, 1998.

[3] M. Arenas, P. Barcelo, R. Fagin, and L. Libkin. Locally consistent transformations and query answering in data exchange. In *Proc. of the 23rd ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS 2004)*, pages 229–240, 2004.

[4] M. Arenas, L. E. Bertossi, and J. Chomicki. Consistent query answers in inconsistent databases. In *Proc. of the 18th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS'99)*, pages 68–79, 1999.

[5] D. Beneventano, S. Bergamaschi, F. Guerra, and M. Vincini. Querying a super-peer in a schema-based super-peer network. In *Proc. of the 3rd VLDB Int. Workshop on Databases, Information Systems and Peer-to-Peer Computing (DBISP2P 2005)*, 2005.

[6] 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.

[7] M. Bouzeghoub and M. Lenzerini. Introduction to the special issue on data extraction, cleaning, and reconciliation. *Information Systems*, 26(8):535–536, 2001.

[8] L. Bravo and L. Bertossi. Logic programming for consistently querying data integration systems. In *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI 2003)*, pages 10–15, 2003.

[9] L. Bravo and L. Bertossi. Disjunctive deductive databases for computing certain and consistent answers to queries from mediated data integration systems. *Journal of Applied Logic – Special Issue on Logic-based Methods for Information Integration*, 3(2):329–367, 2005.

[10] A. Calì, D. Calvanese, G. De Giacomo, and M. Lenzerini. Data integration under integrity constraints. *Information Systems*, 29:147–163, 2004.

[11] A. Calì, D. Calvanese, G. De Giacomo, M. Lenzerini, P. Naggar, and F. Vernacotola. IBIS: Semantic data integration at work. In *Proc. of the 15th Int. Conf. on Advanced Information Systems Engineering (CAiSE 2003)*, pages 79–94, 2003.

[12] A. Calì, D. Lembo, and R. Rosati. On the decidability and complexity of query answering over inconsistent and incomplete databases. In *Proc. of the 22nd ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS 2003)*, pages 260–271, 2003.

[13] 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)*, pages 16–21, 2003.

[14] A. Calì, D. Lembo, R. Rosati, and M. Ruzzi. Experimenting data integration with DIS@DIS. In *Proc. of the 16th Int. Conf. on Advanced Information Systems Engineering (CAiSE 2004)*, volume 3084 of *Lecture Notes in Computer Science*, pages 51–56. Springer, 2004.

[15] D. Calvanese, E. Damaggio, G. De Giacomo, M. Lenzerini, and R. Rosati. Semantic data integration in P2P systems. In *Proc. of the Int. Workshop on Databases, Information Systems and Peer-to-Peer Computing (DBISP2P 2003)*, 2003.

[16] 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 the 9th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2004)*, pages 469–478, 2004.

[17] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. DL-Lite: Tractable description logics for ontologies. In *Proc. of the 20th Nat. Conf. on Artificial Intelligence (AAAI 2005)*, pages 602–607, 2005.

[18] D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, and R. Rosati. Description logic framework for information integration. In *Proc. of the 6th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 2–13, 1998.

[19] D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, and R. Rosati. Data integration in data warehousing. *Int. J. of Cooperative Information Systems*, 10(3):237–271, 2001.

[20] D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Logical foundations of peer-to-peer data integration. In *Proc. of the 23rd ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS 2004)*, pages 241–251, 2004.

[21] D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Y. Vardi. Rewriting of regular expressions and regular path queries. *J. of Computer and System Sciences*, 64(3):443–465, 2002.

[22] D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Y. Vardi. View-based query processing: On the relationship between rewriting, answering and losslessness. In *Proc. of the 10th Int. Conf. on Database Theory (ICDT 2005)*, volume 3363 of *Lecture Notes in Computer Science*, pages 321–336. Springer, 2005.

[23] D. Calvanese, G. De Giacomo, and M. Y. Vardi. Decidable containment of recursive queries. *Theoretical Computer Science*, 336(1):33–56, 2005.

[24] M. J. Carey, L. M. Haas, P. M. Schwarz, M. Arya, W. F. Cody, R. Fagin, M. Flickner, A. Luniewski, W. Niblack, D. Petkovic, J. Thomas, J. H. Williams, and E. L. Wimmers. Towards

heterogeneous multimedia information systems: The Garlic approach. In *Proc. of the 5th Int. Workshop on Research Issues in Data Engineering – Distributed Object Management (RIDE-DOM'95)*, pages 124–131. IEEE Computer Society Press, 1995.

[25] M. A. Casanova, R. Fagin, and C. H. Papadimitriou. Inclusion dependencies and their interaction with functional dependencies. *J. of Computer and System Sciences*, 28(1):29–59, 1984.

[26] 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.

[27] A. K. Chandra and M. Y. Vardi. The implication problem for functional and inclusion dependencies is undecidable. *SIAM J. on Computing*, 14(3):671–677, 1985.

[28] S. S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. D. Ullman, and J. Widom. The TSIMMIS project: Integration of heterogeneous information sources. In *Proc. of the 10th Meeting of the Information Processing Society of Japan (IPSJ'94)*, pages 7–18, 1994.

[29] B. F. Chellas. *Modal Logic: An introduction*. Cambridge University Press, 1980.

[30] J. Chomicki, J. Marcinkowski, and S. Staworko. Computing consistent query answers using conflict hypergraphs. In *Proc. of the 13th Int. Conf. on Information and Knowledge Management (CIKM 2004)*, pages 417–426, 2004.

[31] J. Chomicki, J. Marcinkowski, and S. Staworko. Hippo: a system for computing consistent query answers to a class of SQL queries. In *Proc. of the 9th Int. Conf. on Extending Database Technology (EDBT 2004)*, pages 841–844. Springer, 2004.

[32] I. Clarke, S. G. Miller, T. W. Hong, O. Sandberg, and B. Wiley. Freenet: A distributed anonymous information storage and retrieval system. In *Proc. of the Int. Workshop on Design Issues in Anonymity and Unobservability (DIAU 2000)*, 2000.

[33] O. M. Duschka, M. R. Genesereth, and A. Y. Levy. Recursive query plans for data integration. *J. of Logic Programming*, 43(1):49–73, 2000.

[34] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge*. The MIT Press, 1995.

[35] 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.

[36] R. Fagin, P. G. Kolaitis, and L. Popa. Data exchange: Getting to the core. In *Proc. of the 22nd ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS 2003)*, pages 90–101, 2003.

[37] E. Franconi, G. Kuper, A. Lopatenko, and L. Serafini. A robust logical and computational characterisation of peer-to-peer database systems. In *Proc. of the VLDB International Workshop On Databases, Information Systems and Peer-to-Peer Computing (DBISP2P 2003)*, 2003.

[38] A. Fuxman, E. Fazli, and R. J. Miller. ConQuer: Efficient management of inconsistent databases. In *Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, pages 155–166, 2005.

[39] A. Fuxman, P. G. Kolaitis, R. Miller, and W. C. Tan. Peer data exchange. In *Proc. of the 24rd ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS 2005)*, pages 160–171, 2005.

[40] A. Fuxman and R. J. Miller. First-order query rewriting for inconsistent databases. In *Proc. of the 10th Int. Conf. on Database Theory (ICDT 2005)*, volume 3363 of *LNCS*, pages 337–351. Springer, 2005.

[41] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J. D. Ullman, V. Vassalos, and J. Widom. The TSIMMIS approach to mediation: Data models and languages. *J. of Intelligent Information Systems*, 8(2):117–132, 1997.

[42] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Proc. of*

the 5th Logic Programming Symposium, pages 1070–1080. The MIT Press, 1988.

[43] M. R. Genereseth, A. M. Keller, and O. M. Duschka. Infomaster: An information integration system. In *Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, pages 539–542, 1997.

[44] C. H. Goh, S. Bressan, S. E. Madnick, and M. D. Siegel. Context interchange: New features and formalisms for the intelligent integration of information. *ACM Trans. on Information Systems*, 17(3):270–293, 1999.

[45] G. Grahne and A. O. Mendelzon. Tableau techniques for querying information sources through global schemas. In *Proc. of the 7th Int. Conf. on Database Theory (ICDT'99)*, volume 1540 of *Lecture Notes in Computer Science*, pages 332–347. Springer, 1999.

[46] G. Greco, S. Greco, and E. Zumpano. A logical framework for querying and repairing inconsistent databases. *IEEE Trans. on Knowledge and Data Engineering*, 15(6):1389–1408, 2003.

[47] 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.

[48] L. Grieco, D. Lembo, M. Ruzzi, and R. Rosati. Consistent query answering under key and exclusion dependencies: Algorithms and experiments. In *Proc. of the 14th Int. Conf. on Information and Knowledge Management (CIKM 2005)*, pages 792–799, 2005.

[49] J. Gryz. Query rewriting using views in the presence of functional and inclusion dependencies. *Information Systems*, 24(7):597–612, 1999.

[50] 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)*, pages 505–516, 2003.

[51] J. Heflin and J. Hendler. A portrait of the Semantic Web in action. *IEEE Intelligent Systems*, 16(2):54–59, 2001.

[52] 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.

[53] T. Kirk, A. Y. Levy, Y. Sagiv, and D. Srivastava. The Information Manifold. In *Proceedings of the AAAI 1995 Spring Symp. on Information Gathering from Heterogeneous, Distributed Enviroments*, pages 85–91, 1995.

[54] 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.

[55] 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.

[56] N. Leone, T. Eiter, W. Faber, M. Fink, G. Gottlob, G. Greco, E. Kalka, G. Ianni, D. Lembo, M. Lenzerini, V. Lio, B. Nowicki, R. Rosati, M. Ruzzi, W. Staniszkis, and G. Terracina. The INFOMIX system for advanced integration of incomplete and inconsistent data. In *Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, pages 915–917, 2005.

[57] H. J. Levesque and G. Lakemeyer. *The Logic of Knowledge Bases*. The MIT Press, 2001.

[58] A. Y. Levy. Logic-based techniques in data integration. In J. Minker, editor, *Logic Based Artificial Intelligence*. Kluwer Academic Publisher, 2000.

[59] A. Y. Levy, A. Rajaraman, and J. J. Ordille. Querying heterogenous information sources using source descriptions. In *Proc. of the 22nd Int. Conf. on Very Large Data Bases (VLDB'96)*, 1996.

[60] A. Y. Levy, D. Srivastava, and T. Kirk. Data model and query evaluation in global information systems. *J. of Intelligent Information Systems*, 5:121–143, 1995.

[61] 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.

[62] J. C. Mitchell. The implication problem for functional and inclusion dependencies. *Informa-*

tion and Control, 56:154–173, 1983.

[63] 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.

[64] R. Pottinger and A. Y. Levy. A scalable algorithm for answering queries using views. In *Proc. of the 26th Int. Conf. on Very Large Data Bases (VLDB 2000)*, pages 484–495, 2000.

[65] 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 *Lecture Notes in Computer Science*. Springer, 2000.

[66] A. Tomasic, L. Raschid, and P. Valduriez. Scaling access to heterogeneous data sources with DISCO. *IEEE Trans. on Knowledge and Data Engineering*, 10(5):808–823, 1998.

[67] J. D. Ullman. Information integration using logical views. *Theoretical Computer Science*, 239(2):189–210, 2000.

[68] S. R. Waterhouse, D. M. Doolin, G. Kan, and Y. Faybishenko. Distributed search in P2P networks. *IEEE Internet Computing*, 6(1):68–72, 2002.

[69] B. Yang and H. Garcia-Molina. Designing a super-peer network. In *Proc. of the 19th IEEE Int. Conf. on Data Engineering (ICDE 2003)*, 2003.

[70] Z. Zhang and A. Mendelzon. Authorization views and conditional query containment. In *Proc. of the 10th Int. Conf. on Database Theory (ICDT 2005)*, volume 3363 of *Lecture Notes in Computer Science*, pages 259–273. Springer, 2005.