Weakly-coupled ontology integration of P2P database systems

Zoran Majkić

Dipartimento di Informatica e Sistemistica, University of Roma "La Sapienza"
Via Salaria 113, I-00198 Rome, Italy
majkic@dis.uniroma1.it
http://www.dis.uniroma1.it/ ∼ majkic/

Abstract. We present the basic elements of a new semantics that captures the modular structure of a data-intensive Peer-to-peer (P2P) system, and opens up the possibility of effective query answering techniques. One of the basic characteristics of peers in our approach is that they (possibly) encapsulate an ontology of a data integration system in order to deal with incomplete Web-based information, and to offer a rich ontology interface to their clients. Each peer acts at the same level with respect to the others, with no direct mapping between them, and no unifying structure above them. Based on the above ideas, we define a general framework for P2P systems, and we describe an incremental algorithm for answering conjunctive queries posed to one peer. The contributions to the query answer of any peer are given by the certain answers that such peer provides. We show that the algorithm computes its answers according to a generalized epistemic modal S5 logic, where frames are partitioned according to the structure of peers. This new semantics for P2P systems models a kind of *intensional* mapping that is fundamentally different from the GLAV (Global and Local As View) *extensional* mappings, traditionally adopted in data integration.

1 Introduction

Peer-to-peer systems offer an alternative to traditional client-server systems for some application domains. A P2P system has no centralized schema and no central administration. In P2P systems, every node (peer) of the system acts as both client and server and provides part of the overall information available from an Internet-scale distributed environment.
At first glance, many of the challenges in designing P2P systems seem to fall clearly under banner of the distributed systems community. However, upon closer examination, the fundamental problem in most P2P systems is the placement and retrieval of *data*. Indeed, current P2P systems focus strictly on handling semantic-free, large-granularity requests for objects by identifier (typically a name), 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 large, opaque, and atomic, and whose content is well-described by their name. Moreover, they are limited to caching, prefetching, or pushing of content at the object level, and know nothing of overlap between objects.

These limitations arise because the P2P world is lacking in the areas of semantics, data transformation, and data relationships, yet these are some of the core strengths of the data management community. Queries, views, and integrity constraints can be used to express relationships between existing objects.

References: The first seminal work which introduces the *autoepistemic semantics* for P2P databases, based on known (i.e. certain) answers from the peers is presented by Lenzerini and me in [1] in February 2003, and also referred by the work of Franconi's group in one internal report of University of Trento, [2], six month later (which compares also our approach to theirs). Such introduced Modal logic framework for P2P database systems guarantee also the *decidability* for query answering, non supported by the first-order semantics. But this unique reference in [2] is successively changed, in the identical paper of Franconi, [3], presented for wider scientific community, by the reference to the work [4] presented at the same Workshop in Berlin, September 2003 (but in which is omitted the reference to the seminal work), so that the trace of this original idea was forgotten. Immediately after this seminal work at 'Roman' school of P2P semantics there was two different ideas in how to extend the epistemic semantics of peers (i.e., data integration systems) to the semantics of the mappings between peers:
1. The 'strong' (*extensional*) GLAV (Global or Local As View) mapping which directly extends the data-integration paradigm also for P2P systems [4,5], paraphrased by an *imperative* sentence *'John must know all facts about the "Italian art in the 15'th century" known by Peter*', where 'John' and 'Peter' are two different peers, and
2. The 'weak' (*intensional*) mapping [6,7,8], paraphrased by a *belief*-sentence *'John believes that also Peter knows something about "Italian art in the 15'th century"*'. Such mappings are weaker than internal extensional GLAV mappings of a peer, so that they grantee the independence of peer individuality also in the presence of mappings between peers (for more details see [6]), but they also distinguish the *certain* answers of a peer 'John' from a *possible* answers of 'Peter'.

Shortly, the motivation for the second approach, discussed in this paper, is the following: It is well known that P2P mappings based on GLAV logical mappings between peer ontologies are very *restrictive*: the logic implications used to impose direct semantic relationships between peer ontologies are too much constrictive in the sense that, given such mappings, one is not free to independently change the proper local ontology of some peer: the internal peer mapping and the external mapping with other peers have the *same* (extensional) expressive power so that peer individuality is destroyed. The logic implications from other peers toward the considered peer impose strong requirements for the local peer ontology, and that is a problem for Web semantic P2P integration where different peers are managed *without* a centralized administration. Thus, such ontology integration is not adequate for real large internet based P2P systems.

The main goal for this work i to obtain the more flexible and modular P2P systems, where each peer can be updated without any external constraint. It is obtained by *indirect* (intensional) mappings: any peer chose to publish to the P2P network the proper subset of world concepts (other concepts he can hide for its private usage) for which it is able to give useful answers to user queries, and to map them to the *intensionally equivalent* concepts (views) [6] defined for other peers.

Technical Preliminaries: Usually database mappings, in some logical language , are

given in a schema level where a database schema is defined, shortly as follows:
Let Dom be a countable set of so-called values. Let Att be a countable set of so-called attributes; for each a in Att, the domain dom(a) is a nonempty subset of Dom. Let R be a set of so-called relation symbols, disjoint from Att; each element r of R has a finite arity, and for each r in R, sort(r) is a finite sequence over Att.
Then a *database schema* is a pair $\mathcal{A} = (S_h, S_n)$, where S_h is a subset of R and S_n is a set of closed formulas (constraints) in the sorted first-order language with sorts Att, constant symbols Dom, relational symbols R, and no function symbols.
Rule-based conjunctive queries are composed by a subexpression $r_1(u_1), ..., r_n(u_n)$, which is a *body*, and $q(\mathbf{x})$ which is a *head* of the rule, as follows:
$q(\mathbf{x}) \longleftarrow r_1(u_1), ..., r_n(u_n)$, $n \geq 0$, where r_i are relation symbols in $\mathcal{A}$, q is a relation symbol not in $\mathcal{A}$, u_i are free tuples (i.e., may use either variables or constants). If $v = (v_1, .., v_m)$ then $r(v)$ is a shorthand for $r(v_1, .., v_m)$. Finally, each variable occurring in **x** must also occur at least once in $u_1, ..., u_n$.
If one can find values for the variables of the rule such that the body holds, then one may deduce the head fact: these facts will be called a resulting *view* of the query . We define the total database $\Upsilon = \{a_i | a_i \in A, A \text{ is a database }\}$, that is, Υ is the union of all databases.
The Plan of this work is following: In Section 2 we present the formal definition for weakly-coupled integration of Web based P2P database systems, where each peer is considered as a single Abstract Object Type for a piece of information in Web. In Section 3 we show the general incremental query answering algorithm for this P2P weakly-coupled integration system. In Section 4 we present the a strong closure between the answering algorithm for the initial user query and the semantics of the *rewritten query* over the logical theory obtained by the P2P hybrid language translation.

2 Formal weakly-coupled P2P framework

Dually to the theory of *algebraic specifications* where an Abstract Data Type (ADT) is specified by a set of operations (constructors), the *coagebraic specification* of a class of systems, i.e., Abstract Object Types (AOT), is characterized by a set of operations (destructors) which tell us what can be *observed* out of a system-*state* (i.e., an element of the carrier), and how can a state be transformed to successor state.
We start introducing the class of coalgebras for database query-answering systems. They are presented in an algebraic style, by providing a co-signature. In particular, sorts include one single ”hidden sort”, corresponding to the carrier of the coalgebra, and other ”visible” sorts for inputs and outputs, which are given a fixed interpretation. Visible sorts will be interpreted as sets without any algebraic structure defined on them. Coalgebraic terms, built only over destructors, have for us a precise interpretation as the basic *observations* that one can make on the states of a coalgebra. Input sorts are considered as the set $\mathcal{L}_Q$ of conjunctive queries, $q(\mathbf{x})$, while output sorts are ”valuations”, that is, the set of a resulting ”views”, for each query $q(\mathbf{x})$ over a database $\mathcal{A}$ (considered as a carrier of the coalgebra).

Definition 1. *A co-signature for Database query-answering system is a triple* $\mathcal{D}_\Sigma = (S, OP, [\_])$*, where S, the sorts, OP, the operators, and [_] the interpretation of visible*

sorts are as follows:
1. $S = (X_A, \mathcal{L}_Q, \Upsilon)$*, where* X_A *is the hidden sort (a set of states of a database A),* $\mathcal{L}_Q$ *is an input sort (set of conjunctive queries), and* Υ *is an output sort (set of all views of databases).*
2. OP is set of operations: a method $Next_q : X_A \times \mathcal{L}_Q \to X_A$*, which corresponds to an execution of a next query* $q(\boldsymbol{x}) \in \mathcal{L}_Q$ *in a current state of a database A, such that a database A pass to the next state; and* $Out_Q : X_A \times \mathcal{L}_Q \to \Upsilon$ *is an attribute which returns with an obtained view of a database for a given query* $q(\boldsymbol{x}) \in \mathcal{L}_Q$*.*
3. [_] is a function mapping each visible sort to a non-empty set.
The Abstract Object Type (AOT) for a query-answering system is given by a coalgebra $< \lambda Next_Q, \lambda Out_Q >: X_A \to X_A^{\mathcal{L}_Q} \times \Upsilon^{\mathcal{L}_Q}$ *, of the polynomial endofunctor* $(\_)^{\mathcal{L}_Q} \times \Upsilon^{\mathcal{L}_Q} : Set \to Set$*, where* λ *denotes the lambda abstraction (Curring) for functions of two variables into functions of one variable (*Z^Y *is a set of all functions from Y to Z).*

In object-oriented terminology, the coalgebras just introduced are expressive enough to specify parametric methods and attributes for a database (conjunctive) query answering systems. In what follows , we conceive a peer P_i as a AOT software module characterized by a network ontology G_i expressed in a language $\mathcal{L}_{\mathcal{O}}$ over an alphabet $\mathcal{A}_{\mathcal{G}_i}$. The internal structure of a peer database is hidden to the user, encapsulated in the way that only its logical relational schema $\mathcal{G}_{T_i}$ (global schema if it is a Data integration system [9] $\mathcal{I}_i = (\mathcal{G}_i, \mathcal{S}_i, \mathcal{M}_i)$, where $\mathcal{G}_i = (\mathcal{G}_{T_i}, \Sigma_{T_i})$, Σ_{T_i} are the integrity constraints, $\mathcal{S}_i$ is a source schema and $\mathcal{M}_i$ is a set of generally GLAV mappings between a global schema $\mathcal{G}_{T_i}$ and a source schema $\mathcal{S}_i$) can be seen by users, and is able to respond to the union of conjunctive queries by *certain* answers (that are true in all database instances (models)): we assume that each AOT peer has a unique model or, otherwise, a *canonical (universal)* [10,11] global database, and that responds by *certain* answers.
In order to be able to communicate with other peer P_j in the network $\mathcal{N}$, each peer P_i has also an export-interface module $\mathcal{M}_{EXP}^{ij}$ composed by groups of ordered pairs of conceptually-compatible queries: we define any two queries conceptually-compatible iff they are *intensionally equivalent*; we denote by $q_i \approx_H q_j$ with $Var(head(q_i)) = Var(head(q_j))$. Note that $q_i \approx_H q_j$ does *not* mean that q_i logically implicates q_j or viceversa, as in GLAV mapping definitions.

Definition 2. *The P2P network system* $\mathcal{N}$ *is composed by* $2 \leq N$ *independent peers. Each peer module* P_i *is defined by* $P_i := \langle (\mathcal{G}_i, \mathcal{S}_i, \mathcal{M}_i), \bigcup_{i \neq j \in N} \mathcal{M}_{EXP}^{ij} \rangle$
where $\mathcal{I}_i = (\mathcal{G}_i, \mathcal{S}_i, \mathcal{M}_i)$ *is the encapsulated Data integration system with* $\mathcal{G}_i = (\mathcal{G}_{T_i}, \Sigma_{T_i})$*: only* $\mathcal{G}_{T_i}$*, which is its logical relational schema, is its not-hidden part, and can be seen by users in order to formulate a query.* $\mathcal{M}_{EXP}^{ij}$ *is a (possibly empty) interface to other peer* P_j *in the network, defined as a group of query-connections, denoted by* $(q_{1k}^{ij}, q_{2k}^{ij})$ *where* q_{1k}^{ij} *is a conjunctive query defined over* $\mathcal{G}_{T_i}$*, while* q_{2k}^{ij} *is a conjunctive query defined over the ontology* $\mathcal{G}_{T_j}$ *of the connected peer* P_j *(* $\mid ij \mid$ *denotes the total number of query-connections of the peer* P_i *toward a peer* P_j*) :*

$$\mathcal{M}_{EXP}^{ij} = \{(q_{1k}^{ij}, q_{2k}^{ij}) \mid q_{1k}^{ij} \approx_H q_{2k}^{ij},\ and\ 1 \leq k \leq \mid ij \mid\}$$

In the context of this work we will consider each temporary instance (in a some time t_k) of the P2P database system $\mathcal{N}$ as a particular possible world $w \in \mathcal{W}_N$: the dynamic

changes of any local peer knowledge will result in one other possible world. In what follows we will use one simplified modal logic framework [6] (we will not consider the time as one independent parameter as in Montague's original work [12]) with a model $\mathcal{M}_N = (\mathcal{W}_N, \mathcal{R}_N, S, V)$, where $\mathcal{W}_N$ is the set of possible worlds for a P2P system, $\mathcal{R}_N$ is the accessibility relation between worlds ($\mathcal{R}_N \subseteq \mathcal{W}_N \times \mathcal{W}_N$), S is a non-empty domain of individuals, while V is a function defined for the following two cases:
1. $V : \mathcal{W}_N \times F \rightarrow \bigcup_{n<\omega} S^{S^n}$, with F a set of functional symbols of the language, such that for any world $w \in \mathcal{W}_N$ and a functional symbol $f \in F$, we obtain a function $V(w, f) : S^{arity(f)} \rightarrow S$.
2. $V : \mathcal{W}_N \times P \rightarrow \bigcup_{n<\omega} \mathbf{2}^{S^n}$, with P a set of predicate symbols of the language and $\mathbf{2} = \{t, f\}$ is the set of truth values (true and false, respectively), such that for any world $w \in \mathcal{W}_N$ and a predicate symbol $p \in P$, we obtain a function $V(w, p) : S^{arity(p)} \rightarrow \mathbf{2}$, which defines the extension $[p] = \{\mathbf{a} | \mathbf{a} \in S^{arity(p)} \text{ and } V(w, p)(\mathbf{a}) = t\}$ of this predicate p in the world w.
The extension of an expression α, w.r.t. a model $\mathcal{M}_N$, a world $w \in \mathcal{W}_N$ and assignment g is denoted by $[\alpha]^{\mathcal{M}_N, w, g}$. Thus, if $c \in F \bigcup P$ then for a given world $w \in \mathcal{W}_N$ and the assignment function for variables g, $[c]^{\mathcal{M}_N, w, g} = V(w, c)$, while for any formula A, $\mathcal{M}_N \vDash_{w,g} A \equiv ([A]^{\mathcal{M}_N, w, g} = t)$, means 'A is true in the world w of a model $\mathcal{M}_N$ for assignment g'. Montague defined the *intension* of an expression α by:
$[\alpha]^{\mathcal{M}_N, g}_{in} =_{def} \{w \mapsto [\alpha]^{\mathcal{M}_N, w, g} \mid w \in \mathcal{W}_N\}$,
i.e., as graph of the function $[\alpha]^{\mathcal{M}_N, g}_{in} : \mathcal{W}_N \rightarrow \bigcup_{w \in \mathcal{W}_N} [\alpha]^{\mathcal{M}_N, w, g}$.
One thing that should be immediately clear is that intensions are more general that extensions: if the intension of an expression is given, one can determine its extension with respect to a particular world but not viceversa, i.e., $[\alpha]^{\mathcal{M}_N, w, g} = [\alpha]^{\mathcal{M}_N, g}_{in}(w)$.
In particular, if c is a non-logical constant (individual constant or predicate symbol), the definition of the extension of c is, $[c]^{\mathcal{M}_N, w, g} =_{def} V(w, c)$.
Carnap suggested that the intension of an expression is nothing more than all the varying extensions the expression can have. Based on this we define that two expressions (or concepts) α, β are *intensionally equivalent* as follows:

Definition 3. *Any two expressions, α, β, are intensionally equivalent, denoted by $\alpha =_{in} \beta$, if and only if* $\bigcup_{w \in \mathcal{W}_N} [\alpha]^{\mathcal{M}_N, g}_{in}(w) = \bigcup_{w \in \mathcal{W}_N} [\beta]^{\mathcal{M}_N, g}_{in}(w)$
so that for any two conjunctive queries, $q_i(x)$, $q_j(x)$ over peers P_i, P_j with their epistemic modal operators K_i, K_j (used to obtain certain answers from peers) respectively, we define: $q_i(x) \approx_H q_j(x)$ *if and only if* $K_i\, q_i(x) =_{in} K_j\, q_j(x)$.

3 Query answering

Conjunctive queries to $\mathcal{N}$ are posed in term of the ontology $\mathcal{G}_{T_i}$ of the peer P_i, and are expressed in a query language $\mathcal{L}_{\mathcal{O}}$ over an alphabet $\mathcal{A}_{\mathcal{G}_i}$. There are two ways to formally consider such *certain* query answering:
AOT Framework: We can enrich the global schema $\mathcal{G}_{T_i}$ by a new unary predicate $Val(\_)$ such that $Val(c)$ is true if $c \in Dom$ is a constant of the local ontology of this peer, and Val(c) is false if c is a constant of some Skolem function introduced for virtual predicates of a global schema, in order to overcame incomplete information in source

data of peer database which encapsulate a data integration system (for example, when we force foreign key constraints in a global schema we use the existentially quantified rules: such quantifiers are eliminated by introducing Skolem functions for these existentially quantified attributes).
The AOT module of a peer, transforms every original query $q_C(\mathbf{x})$ over its global schema, where $\mathbf{x} = x_1, .., x_k$ is a non empty set of variables, into a *lifted query*, denoted by q, such that $q := q_C(\mathbf{x}) \wedge Val(x_1) \wedge ... \wedge Val(x_k)$. With Such transformation AOT eliminates all tuples with Skolem constants from the set of answers.
The universal (canonical) database $can(\mathcal{I}, \mathcal{D})$, [10,11], of the encapsulated Data integration system with the source database $\mathcal{D}$, has the interesting property of faithfully representing all legal databases. In practice we do not use this canonical database in order to give the answer to the query, and we use a *query rewriting technics under constraints* in data integration systems [10] to submit the rewritten query directly to source databases.
Modal Logic Framework: The autoepistemic semantics for P2P mappings was first time introduced in [1]. Following Levesque [13] queries should be formulas in an epistemic modal logic. On this view, integrity constraints are *modal sentences* and hence are formally identical to a strict subset of permissible database queries. They defined a first-order modal language $KFOPCE$ with a single universal modal operator $\Box \equiv K$ (for "know"). The resulting modal logic is weak S5 (also known K45): In fact given a single data integration system, the possible worlds are legal databases of this data integration system each one connected to all other. Thus a modal query formula at some world $\Box q_C$ is a believed (certain) iff q_C is true at all possible worlds (i.e., legal databases) accessible from that world.

We will generalize such approach to the P2P network of disjoint peers (data integration systems) in "locally modular" way, such that, by introducing new or by eliminating some old peer, the frame of this modal logic is only locally changed in very bounded way. The resulting modal logic is also week S5 composed by completely *disjoint partitions* of worlds: each partition corresponds to one particular peer and is composed by all legal databases (possible worlds) of that peer, connected each one with all others.
Such modularization generalizes the modal logic framework: each peer P_i can be seen as an independent AOT *agent*, whose semantics is defined by a particular ("local") epistemic modal logic which express its knowledge by a *local* modal operator K_i (for "peer P_i knows").
As usual, the semantics of formulas of a modal logic is described by means of the notation $\mathcal{M} \vDash_{\mathcal{A},\, g} q$ with the meaning: " q is true in the model $\mathcal{M}$, at the point $\mathcal{A}$ and for the assignment function g ". The semantics of the modal operators $\Box$ and $\Diamond$ are:
$\mathcal{M} \vDash_{\mathcal{A},\, g} \Box q$ iff $\mathcal{M} \vDash_{\mathcal{A}',\, g} q$ for every $\mathcal{A}'$ in $\mathcal{W}$ such that $\mathcal{A}\mathcal{R}\mathcal{A}'$.
$\mathcal{M} \vDash_{\mathcal{A},\, g} \Diamond q$ iff there exists a $\mathcal{A}'$ in $\mathcal{W}$ such that $\mathcal{A}\mathcal{R}\mathcal{A}'$ and $\mathcal{M} \vDash_{\mathcal{A}',\, g} q$.
A formula q is said to be *true in a model* $\mathcal{M}$ if $\mathcal{M} \vDash_{\mathcal{A},\, g} q$ for each g and $\mathcal{A} \in \mathcal{W}$.
A formula is said to be *valid* if it is true in each model.
Let $\mathcal{A}$ and $\mathcal{B}$ be any two databases, considered as possible worlds (points) in the Kripke model $\mathcal{M}$, with φ_A a set of all integrity constraints for a database $\mathcal{A}$ and ϕ_B a set of all integrity constraints for a database $\mathcal{B}$. Thus, in a modal logic we have that for any assignment function g, $\mathcal{M} \vDash_{\mathcal{A},\, g} \varphi_A$ and $\mathcal{M} \vDash_{\mathcal{B},\, g} \phi_B$. Any 'strong' (or direct)

mapping between peer-databases, instead, which uses a certain (known) answers from $\mathcal{A}$ to $\mathcal{B}$, can be expressed by in the following way (in a Gabbay-style rule) [14]:
" $\mathcal{M} \models_{\mathcal{A},\, g} \Box q_A(\mathbf{x})$ implies $\mathcal{M} \models_{\mathcal{B},\, g} \Box q_B(\mathbf{x})$ "
These are metatheoretic considerations, they are not formulas of the standard modal logic, and are a "local" one: we need that some formulas be true at exactly *one* point in any model of P2P logic theory (a particular peer-database). Thus, in order to define mappings between databases with certain (known) answers of these databases, we need a "hybrid" extension [15] of the ordinary modal logic, able to capture each particular modal operator K_i of a peer P_i.
We can do this by introducing a second sort of atomic formula: *nominals*. Syntactically these will be ordinary atomic formulas, but they will have an important *semantic* property: nominals will be true at exactly *one* point in any model of the modal logic; nominals "name" this point by being true there and nowhere else.
The new modal operator, @, for this hybrid logic enables to "retrieve" worlds. More precisely, a formula of the form $@_x\varphi$ is an instruction to *move* to the world labelled by the variable x and evaluate φ there.
Any mapping, which uses a certain (known) answers from $\mathcal{A}$ to $\mathcal{B}$, expressed in a Gabbay-style rule: " $\mathcal{M} \models_{\mathcal{A},\, g} \Box q_A(\mathbf{x})$ implies $\mathcal{M} \models_{\mathcal{B},\, g} \Box q_B(\mathbf{x})$ " can now be directly translated into hybrid modal logic formula: let i be the nominal used for the peer-database $\mathcal{A}$, and k be the nominal used for the peer-database $\mathcal{B}$, then

- $@_i \Box q_A(\mathbf{x}) \Rightarrow @_k \Box q_B(\mathbf{x})$, so that a particular modal operator of each peer P_i can be defined by $K_i = @_i \Box$.

But such 'strong' mapping between peers, which imposes that the certain answers to the query $q_B(\mathbf{x})$ of the peer $\mathcal{B}$ has to contain all certain answers of the query $q_A(\mathbf{x})$ of the peer $\mathcal{A}$, is *to much restrictive* for real P2P applications. Because of that we will use only 'weakly-coupled' *intensional* mappings [6] between peers.
Query answering algorithm: The general scenario for query answering in a P2P network system $\mathcal{N}$ composed by $2 \leq N$ peers $P_i := \langle(\mathcal{G}_i, \mathcal{S}_i, \mathcal{M}_i),\ \bigcup_{i \neq j \in N} \mathcal{M}^{ij}_{EXP}\rangle$ can be described as follows:
Given an initial conjunctive user query $q_C(\mathbf{x})$, where $\mathbf{x} = x_1, .., x_k$ is a non empty set of variables, over the global schema of some selected peer P_k, we denominate "answering pair" $(q_k(\mathbf{x}), P_k)$, where $q_k(\mathbf{x}) = q_C(\mathbf{x})$ is the query rewritten for the k-th peer in the network. Let denote by $\mathcal{L}_{ans}$ the list of all answering pairs for a given user query: intuitively such list contains a pair $(q_C(\mathbf{x}), P_k)$ from whom we can obtain the *certain* answers, and other (reformulated-query, peer)-pairs which can give the *possible* answers known by that peers. We denote by q^{P_i} the known answer of the i-th peer to the user query. The maximal answer $\mathcal{A}_{ns}$ to the user query is the union of the certain answers of the interrogated peer P_k, and the possible answers of *all* other answering pairs in the P2P network $\mathcal{N}$.
This answering list $\mathcal{L}_{ans}$ have to be generated dynamically during the answering process, where at each step, when some peer is selected as a candidate in order to give an answer, its network interface is used to verify if there is some other peer, connected to it, able to answer to such a query.
Let us describe the general query answering algorithm Alg_{P2P} for *pure* [6] P2P system:

Definition 4. *Let K be a finite natural number which limits the partial answers for every peer during the process of query answering of a P2P system (we consider also that one peer can be indicated by other peers more than one time in order to answer to the query).*

1. *Set $\mathcal{A}_{ns} = \{\}$ and $\mathcal{L}_{ans} = (q_k(\boldsymbol{x}), P_k)$, where $q_k(\boldsymbol{x}) = q_C(\boldsymbol{x})$ is the original user query formalized over the global scheme $\mathcal{G}_{T_k}$ of the peer P_k.*
2. *Take from $\mathcal{L}_{ans}$ the first answering pair $(q_i(\boldsymbol{x}), P_i)$ which didn't give yet the answer to the query. If there is not such a pair the answering process is finished, with the maximal answer in $\mathcal{A}_{ns}$. Otherwise calculate the known answer of this peer; obtained tuples are added to $\mathcal{A}_{ns}$.*
3. *Consider only interface group $\mathcal{M}^{ij}_{EXP} = \{(q^{ij}_{1k}, q^{ij}_{2k}) \mid q^{ij}_{1k} \approx_H q^{ij}_{2k},\ \ 1 \leq k \leq \mid ij \mid\}$, of the current peer P_i toward the peer P_j, which is not inserted more than K times in the list $\mathcal{L}_{ans}$. If such interface group does not exists go to the point 2.*
   Otherwise, build the local mapping system in the following way:
   (a) *for each query connection $(q^{ij}_{1k}, q^{ij}_{2k}) \in \mathcal{M}^{ij}_{EXP}$, define a 'completed' [6] predicate r^{ij}_k and the GAV mapping $\mathcal{M}_{GAV}$, by inserting: $(q^{ij}_{2k} \Longrightarrow r^{ij}_k) \in \mathcal{M}_{GAV}$.*
   (b) *Try now to rewrite the query $q_i(\boldsymbol{x})$, in terms of the set of views $S_V = \{\, q^{ij}_{1k} \mid (q^{ij}_{1k}, q^{ij}_{2k}) \in \mathcal{M}^{ij}_{EXP}\}$ of the current peer [16]. If it is not possible, go to the next interface group of the current peer (point 3); Otherwise, define the query $q_G(\boldsymbol{x})$ by replacing each original virtual predicate q^{ij}_{1k} in S_V by, intensionally equivalent to it, 'completed' predicate r_k, $1 \leq k \leq \mid ij \mid$.*
   (c) *Rewrite now by unfolding, using $\mathcal{M}_{GAV}$ mappings in (b), the query $q_G(\boldsymbol{x})$ into the query $q_j(\boldsymbol{x})$ over the logical scheme $\mathcal{G}_{T_j}$ of the peer P_j, and insert this new answering pair $(q_j(\boldsymbol{x}), P_j)$ in the list $\mathcal{L}_{ans}$.*
   Go to the next interface group of the current peer (point 3).

Note that such algorithm is parameterized by $K = 1, 2, \ldots$ and that it is monotone with respect to K: the number of tuples, in the answer to the query, grows by increasing its value, but the total time to reach the end of this process grows also. In the case when K is infinite the process described by Alg_{P2P} theoretically may not terminate.

Theorem 1 *Let $\mathcal{N}$ be a finite P2P network system and q_C be a conjunctive query. Then the query answering algorithm Alg_{P2P} terminates and the result of this query is a finite union of certain answers in $\mathcal{A}_{ns}$.*

Note that, given the same conjunctive query q_C over initially different peers, will generally be obtained different maximal answers on a given P2P network system: the answers to a query are topology-dependent, with the following important feature:
Incremental query answering: the process of answering to queries can be also controlled by user, in a sense that, in the step 2 of the algorithm we can introduce the possibility that the user may interrupt the execution of the algorithm: the system may present partial results after each peer which have (partially) answered to the original user query, and if the user is satisfied by obtained results, he can interrupt the process. In Web applications this is usually an important requirement, because the time used in order to obtain maximal answers can be considerably long.

4 Model and Semantics for P2P Network System

Let us define a virtual network database schema $\mathcal{N}_{DB}$ (without integrity constraints), for the given P2P network $\mathcal{N}$ composed by N peers P_i:

$\mathcal{N}_{DB} = \biguplus_{P_i \in \mathcal{N}} \bigcup_{\mathcal{M}^{ij}_{EXP} \in \mathcal{P}_i} \{ r^{ij}_{ik} \mid Var(r^{ij}_{ik}) \triangleq head(q^{ij}_{1k}), (q^{ij}_{1k}, q^{ij}_{2k}) \in \mathcal{M}^{ij}_{EXP} \}$
where r^{ij}_{ik} are the new 'completed' [6] relation symbols (predicates) of this database and " $\biguplus$ " is the disjoint union operation.

Definition 5. *The hybrid model $\mathcal{M} = (\mathcal{W}, \mathcal{R}, \mathcal{S}, \mathcal{V}, \mathcal{V}_N)$ for each P2P network system $\mathcal{N}$ is a particular S5 model where:*

- *For each peer $\mathcal{P}_i$ we define one disjoint partition, $\mathcal{W}_i$, composed by the following possible worlds (points):*
  1. One point for a Data integration system encapsulated by this peer.
  2. A point for each legal database w.r.t this data integration system.
  Thus, the set of all worlds (or points) $\mathcal{W}$ is defined by: $\mathcal{W} = \{\mathcal{N}_{DB}\} \biguplus_{i \in N} \mathcal{W}_i$, where $\mathcal{N}_{DB} = \mathcal{V}_N(n_N)$ (denotation of n_N) is the unique network database world.
- *$\mathcal{R}$ is a binary relation of "accessibility" on a set $\mathcal{W}$, defined as follows:*
  for each peer $P_i := \langle (\mathcal{G}_i, \mathcal{S}_i, \mathcal{M}_i), \bigcup_{i \neq j \in N} \mathcal{M}^{ij}_{EXP} \rangle$, we create a disjoint partition, $\mathcal{R}_i$, of the global binary relation $\mathcal{R} = (\mathcal{N}_{DB}, \mathcal{N}_{DB}) \biguplus_{i \in N} \mathcal{R}_i$:
  $(\mathcal{G}_i, G_k) \in \mathcal{R}_i$, $(G_k, \mathcal{G}_i) \in \mathcal{R}_i$ and $(G_j, G_k) \in \mathcal{R}_i$,
  where G_k, G_j , $j, k = 1, 2, ..$ are possible models (legal databases instances of data integration system $(\mathcal{G}_i, \mathcal{S}_i, \mathcal{M}_i)$). Each G_k can be seen as a logical theory also, composed by only ground terms.
- *$\mathcal{S} \biguplus_{i \in N} S_i$ is the disjoint union of non-empty domains S_i of peer-individuals.*
- *$\mathcal{V} = \{\mathcal{V}_i \mid i \in N\}$ is a set of functions: $\mathcal{V}_i$ assigns to each pair consisting of a n-place predicate constant P and of an element $\mathcal{A} \in \mathcal{W}_i$ a function $\mathcal{V}_i(P, \mathcal{A})$ from S^n_i to $\{1, 0\}$.*
- *$\mathcal{V}_N$ is a function which assigns to each nominal i, a point $P_i \in \mathcal{W}$.*

So we obtain the modularization of the frame of the global Kripke model
$M = (\mathcal{W}, \mathcal{R}, \mathcal{S}, \mathcal{V}, \mathcal{V}_N) = (\{\mathcal{N}_{DB}\}, (\mathcal{N}_{DB}, \mathcal{N}_{DB}), \mathcal{S}, \mathcal{V}, \mathcal{V}_N) \biguplus_{i \in N} M_i$,
where $M_i = (\mathcal{W}_i, \mathcal{R}_i, \mathcal{S}_i, \mathcal{V}_i, \mathcal{V}_N)$ is a disjoint portion ("local" Kripke model for a peer P_i) of the global Kripke model, with a "local" modal operator K_i of a peer P_i.
Now we are able to translate directly the mappings of a P2P network system $\mathcal{N}$ into ordinary syntax of this hybrid modal logic.

Definition 6. *(Hybrid modal logic translation)*
For every peer $P_i := \langle (\mathcal{G}_i, \mathcal{S}_i, \mathcal{M}_i), \bigcup_{i \neq j \in N} \mathcal{M}^{ij}_{EXP} \rangle$ in a P2P network system $\mathcal{N}$, we do as follows:

- *Internal, encapsulated, structure of a peer: all assertions of its integrity constraints Σ_{T_i}, all assertions in its mapping $\mathcal{M}_i$ and all facts (ground terms) of a "local" logical theory of this data integration system are prefixed by the modal operator $@_i$, where the point (database) $\mathcal{G}_i = V_N(i)$ is the denotation of i for a peer P_i.*
- *Network interface mappings: for every query-connection $(q^{ij}_{1k}, q^{ij}_{2k}) \in \mathcal{M}^{ij}_{EXP}$, between peer P_i and P_j, i.e., $q^{ij}_{1k} \approx_H q^{ij}_{2k}$, we define the pair of closed sentences based on logical implications:*

$\forall \boldsymbol{x}(@_i \Box q_{1k}^{ij}(\boldsymbol{x}) \Longrightarrow @_{n_N} r_{ik}^{ij}(\boldsymbol{x}))$ *, i.e.* $K_i\ q_{1k}^{ij}(\boldsymbol{x}) \Longrightarrow r_{ik}^{ij}(\boldsymbol{x})$
$\forall \boldsymbol{x}(@_j \Box q_{2k}^{ij}(\boldsymbol{x}) \Longrightarrow @_{n_N} r_{ik}^{ij}(\boldsymbol{x}))$ *, i.e.* $K_j\ q_{2k}^{ij}(\boldsymbol{x}) \Longrightarrow r_{ik}^{ij}(\boldsymbol{x})$
where $r_{ik}^{ij} \in \mathcal{N}_{DB}$*, and* $\boldsymbol{x}$ *is a list of query variables.*

The extension of network virtual predicates r_{ik}^{ij} is the *exact* contribution of all peers by their proper extension of certain answer in the left side of formulae above: we denominate them by 'completed' predicates. Notice that the translated P2P mappings, from the intensionally-based equivalence, $\approx_H$, of the set of query-connections contained in ADT's of peers, into logic formulae are not reachable by traditional GLAV mappings used in Data integration/exchange systems: in fact no one of these two paired queries (for two different peers)implicates other one.
Thus, the weakly-coupled P2P system has its *proper P2P mapping semantics*, expressed exclusively by implication from a query over a given peer to the, intensionally-equivalent to it, the 'completed' predicate in a network database $\mathcal{N}_{DB}$. It is important to underline that this network database $\mathcal{N}_{DB}$ **is not** any kind of Global ontology of a P2P system and that is not object for any user query: its rule is technical one only, and its parts are locally and dynamically reconstructed by the query answering algorithm only. We can give the main result about general framework for P2P network systems.

Theorem 2 *(General Framework for P2P Network Systems) The logical theory of each P2P network system* $\mathcal{N}$ *can be formalized by closed sentences of the Hybrid sublanguage* $\mathcal{H}(@)$ *for the S5 normal modal logic, enriched by nominals* $n \in NOM$ *and by satisfaction modal operators* $@_n$*.*

Now we are ready to correlate the algorithm Alg_{P2P} with the Hybrid modal semantics of P2P systems:

Theorem 3 *Let* $\mathcal{N}$ *be a P2P system,* q_C *the initial user query formalized over the peer* P_i *and* $q_{P2P} = \bigvee_{n \in \mathcal{N}\ \&\ 1 \leq j \leq K} @_n \Box q_{nj}$ *, where* $q_{i1} = q_C$ *and* q_{nj}*,* $1 \leq j \leq K$ *are the queries rewritten (by the algorithm* Alg_{P2P}*) for the peer* P_n *(*$n \in \mathcal{N}$*), then the answer of the logical theory, obtained by the hybrid language translation of this P2P system, to such query* q_{P2P} *is equal to the answer obtained by the algorithm* Alg_{P2P} *for the initial user query* q_C*.*

Notice that in such semantics of P2P systems, the strong closure relationship between semantics and query rewriting algorithms in Data integration systems *does not exist*. Indeed, such strong closure for Data integration system is based on the fact that *there exists* the global schema (with integrity constraints) as a base for definitions of user query: The strong closure means that the answers to the rewritten queries over source databases coincide to the certain tuples, for the original user query, obtained from the canonical database of the global schema.
In the P2P systems such global schema does not exist, and is not possible to define a query over it: each user query is defined over a schema *of some particular* peer, which is not a global schema of a P2P system. Thus, differently from strong closure in Data integration systems, the answer to such user query is bigger than certain answers of query-interrogated peer: this peer are not able to respond for other peers; it is task of a

query-agent to reformulate original user query to other, 'connected' by intensional mapping by the K-parameterized algorithm Alg_{P2P}, peers in order to obtain more (possible) information.
Here we have an other kind of the question: which is the value of K for which, for a given P2P system interconnections, we can obtain the *maximal possible answer* to the given query. The theorem above tells us that, fixed any initial user query q_C over the peer P_i, for different values of the parameter K we obtain *different rewritten queries* over the logical theory obtained by the hybrid language translation of this P2P system: the valid tuples (of a type defined by the head of the initial user query q_C) of each one of this rewritten query q_{P2P} are equal to the answer given by the algorithm Alg_{P2P}. Thus there is a strong closure between the answering algorithm for the initial user query q_C and the semantics of the *rewritten query* q_{P2P} over the logical theory obtained by the P2P hybrid language translation.

5 Conclusion

As this paper shows, the problem of answering queries in P2P network systems raises a multitude of challenges, ranging from theoretical foundations to considerations of a more practical nature. The algorithms for answering queries using views are already incorporated into a number of data integration systems with integrity constraints, and we consider that such technics can be "locally" used in order to obtain certain answers from a single peer. The difficulties basically arise because of the need of dealing with incomplete information so that, by encapsulation of database integration system into each peer, we obtain an adequate answer for a Web based reach-ontology applications.
We have shown that the nature of P2P mappings between peers has a different semantics w.r.t the general GLAV mappings in data integration (or data exchange) systems: it is based on the *intensional semantics*.
The Modular AOT view of peers is also adequate for other engineering challenges.
The *disjoint composition*, indexed by each peer P_i in $\mathcal{N}$, of the frame of the global Kripke model $\mathcal{M} = (\mathcal{W}, \mathcal{R}, \mathcal{S}, \mathcal{V}, \mathcal{V}_N)$, is premise for the pure *modular development and maintenance* of the P2P systems:
1. **Conservative upgrading**: When we add a new peer, we simply add a new disjoint partition in the frame of the model $\mathcal{M}$ (the new disjoint part of the network database $\mathcal{N}_{DB}$ is implicitly added).
2. **Local updates**: When we modify some peer, we modify only its disjoint partition in the frame of the model $\mathcal{M}$ (its disjoint part of the network database $\mathcal{N}_{DB}$ is automatically modified).
3. **Preserving integrity**: When we eliminate a preexisting peer, we simply eliminate its disjoint partition in the frame of the model $\mathcal{M}$ (its disjoint part of the network database $\mathcal{N}_{DB}$ is automatically eliminated).

These features are the consequences that a *general* hybrid Kripke model is the disjoint union of *local* (for every given peer P_i) hybrid Kripke models and the unique Network database world (there is no any constraint over $\mathcal{N}_{DB}$, so it has a unique model whose extension is a union of information of all peers to its 'completed predicates). Thus, in this framework, each peer can be considered as the local hybrid model with

its particular local modal operator $K_i = @_i\Box$ (for "peer P_i knows"). The P2P design problem is often treated as a problem of *search* through a set of peer configurations: in each configuration, we need to determine wether the workload queries anticipated on some particular peer can be answered using its interface module to other peers, and estimate the cost of the configuration. In particular, this raises to challenge of developing *incremental* algorithms for answering queries. It is immediate to verify that the technique can be easily adapted to deal with the case of *unions* of conjunctive queries.
This research is partially supported by the project SEWASIE-IST-2001-3425. The author wishes to thank Maurizio Lenzerini for his support.

References

1. M.Lenzerini and Z. Majkić. General framework for query reformulation. *Semantic Webs and Agents in Integrated Economies, D3.1, IST-2001-34825, February,*, 2003.
2. E.Franconi, G.Kuper, A.Lopatenko, and L.Serafini. A robust logical and computational characterization of peer-to-peer data systems. *Technical Report DIT-03-051, University of Trento, Italy, September*, 2003.
3. E.Franconi, G.Kuper, A.Lopatenko, and L.Serafini. A robust logical and computational characterization of peer-to-peer data systems. *Proc. of the Int. Workshop On Databases, Inf.Systems and P2P Computing, Berlin, Germany, September*, 2003.
4. D.Calvanese, E.Damaggio, G. De Giacomo, M.Lenzerini, and R.Rosati. Semantic data integration in p2p systems. *Proc. of the Int. Workshop On Databases, Inf.Systems and P2P Computing, Berlin, Germany, September*, 2003.
5. D.Calvanese, G. De Giacomo, M.Lenzerini, and R.Rosati. Logical foundations of peer-to-peer data integration. *PODS 2004, June 14-16, Paris, France*, 2004.
6. Z. Majkić. Intensional mapping in peer-to-peer database systems. *Notes in http://www.dis.uniroma1.it/ ∼ majkic/*, 2004.
7. Z. Majkić. Weakly-coupled p2p system with a network repository. *6th Workshop on Distributed Data and Structures (WDAS'04), July 5-7, Lausanne, Switzerland*, 2004.
8. Z. Majkić. Massive parallelism for query answering in weakly integrated p2p systems. *Workshop GLOBE 04, August 30-September 3,Zaragoza, Spain*, 2004.
9. Maurizio 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.
10. A.Calì, D.Calvanese, G.De Giacomo, and M.Lenzerini. Data integration under integrity constraints. In *Proc. of the 14th Conf. on Advanced Information Systems Engineering (CAiSE 2002)*, pages 262–279, 2002.
11. 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)*, 2003.
12. R.Montague. Formal philosophy. selected papers of richard montague. *in R.Thomason (editor), Yale University Press, New Haven, London*, pages 108–221, 1974.
13. Hector J. Levesque. All I know: a study in autoepistemic logic. *Artificial Intelligence*, 42:263–310, 1990.
14. D.Gabbay. An irreflexivity lemma. *in U.Monnich (ed.) Aspects of Philosophical Logic, Riedel*, pages 67–89, 1981.
15. P.Blackburn. Representation, reasoning, and relational structures: a hybrid logic manifesto. *Methods for Modalities 1, Logic Journal of the IGPL*, 8:339–365, 2000.
16. A.Levy, A.Mendelzon, and Y.Sagiv. Answering queries using views. *Proc. 14th ACM Symp. on Principles of Database Systems*, pages 95–104, 1995.