International Journal of Computer Science and Applications

Vol. XX No. XX, pp. XXX - XXX, 20XX

Semantics-enabled Policies for Super-Peer Data Integration and Protection

Yuh-Jong Hu and Win-Nan Wu and Jiun-Jan Yang
ENT Lab., Dept. of Computer Science
National Chengchi University
Taipei, Taiwan, 11605
hu@cs.nccu.edu.tw, {99753505,98753036}@nccu.edu.tw

Extending from the previous semantic privacy-preserving model, we propose a wide-scale super-peer data integration and protection architecture. Any user from a super-peer domain can contribute new data, schema, or even mappings for other super-peer domains to integrate the information. Each super-peer domain is essentially a mediator-based data integration system, where an agent at the super-peer performs semantic local mappings to manage a set of its local peers endowed with shareable relational data sources. Semantic global mappings are also possible from the current super-peer to interlink with other super-peers located in their super-peer domains. A super-peer is the only place, at the virtual platform (`VP`), where an agent can empower the data integration and access control services for a super-peer domain. Through the semantics-enabled privacy protection policies, authorized view-based queries posed to a super-peer can enable the data integration without losing a user's privacy. The ontology mapping and merging algorithm with a local-as-view (LAV) source description that creates a global ontology schema in a super-peer by integrating multiple local ontology schemas for data integration. The perfect rules integration of datalog rules enforces the data query and protection services. Finally, using a global-local-as-view (GLAV) for global semantic mappings among super-peers, we have a greater flexibility of data integration and protection in the super-peer architecture.

Keywords: Semantics-enabled policies, super-peer data integration, privacy protection, ontology and rule

1. Introduction

Large enterprises spend a great deal of time and money on data (or information) integration [Bernstein and Haas, 2008]. Data integration is the problem of combining data from autonomous and heterogeneous sources, and providing users with a unified view of these data through so called global (or mediated) schema. The global schema, which is a reconciled view of the information, that provides query services to end users. The design of a data integration system is a very complex task and includes several different issues: heterogeneity of the data sources, relation between the global schema and the data sources, limitations on the mechanisms for accessing the sources, and how to process queries expressed on the global schema,

etc [Calvanses et al., 2002].

We face a data request for a tremendous amount of heterogeneous and scalable data sources on the web. A peer data management system (PDMS) inherits the spirit of `PAYGO` approach that enables a wide-scale data integration [Franklin et al., 2005] [Madhavan et al., 2007]. In a PDMS, each peer exports data in terms of its own schema, and information integration is achieved by establishing mappings among the various peer schemas. In the super-peer network architecture, we group a set of peers into a super-peer domain and organize them into a two-level architecture. In the lower level, called the peers, and in the upper one, called the super-peer [Beneventano et al., 2007]. More precisely, a peer integrates data sources into a local ontology. A super-peer contains a data integration system, which integrates these local peers' ontologies into a global ontology through ontology mapping, alignment, and merging. Therefore, a traditional data integration system can be viewed as a special case of a PDMS.

Three approaches have been proposed to model a set of *source descriptions* that specify the semantic mapping between the source schema and the global schema. The first one, called global-as-view (GAV), requires that the each concept in the global schema is expressed in terms of query over the data sources. The GAV deals with the case when the stable data source contains details not present in the global schema so it is not used for dynamically adding or deleting data sources.

The second one, called local-as-view (LAV), requires the global schema to be specified independently from the sources, and the source descriptions between the stable global schema, such as ontology and the dynamic data sources are established by defining each concept in the data sources as a view over the global schema [Calvanese and Giacomo, 2005] [Lenzerini, 2002]. LAV descriptions handle the case in which the global schema contains details that are not present in every data sources.

The third one, called global-local-as-view (GLAV), a source description that combines the expressive power of both GAV and LAV, allows flexible schema definitions independent of the particular details of the data sources [Friedman et al., 1999] [Nash and Deutsch, 2007]. The data integration system uses these different source descriptions to reformulate a user query into a query over the source schemas. However, data integration is hampered by legitimate and widespread privacy concerns, so it is critical to develop a technique that enables the integration and sharing of data without losing a user's privacy [Clifton et al., 2004].

Privacy protection policies represent a long-term promise made by an enterprise to its users and are determined by business practices and legal concerns. It is undesirable to change an enterprise's promises to customers every time an internal access control rule changes. If possible, we should allow the integration of Platform for Privacy Preferences (P3P) and Enterprise Privacy Authorization Language (EPAL) policies to provide accountable and transparent information processing for data owners to revise their data usage permissions [Antón et al., 2007].

Although many organizations post online privacy policies, they must realize

that simply posting a privacy policy on their Web sites does not guarantee true compliance with existing legislation. Following the OECD's Fair Information Principles (FIPs)[a], an organization should provide the norms of personal information processing for its data collection, retention, use, disclosure, and destruction. An organization must also be accountable for its information possession and should declare the purposes of information usage before collection. Moreover, an organization should collect personal information with an individual's consent and disclose personal information only for previously identified purposes.

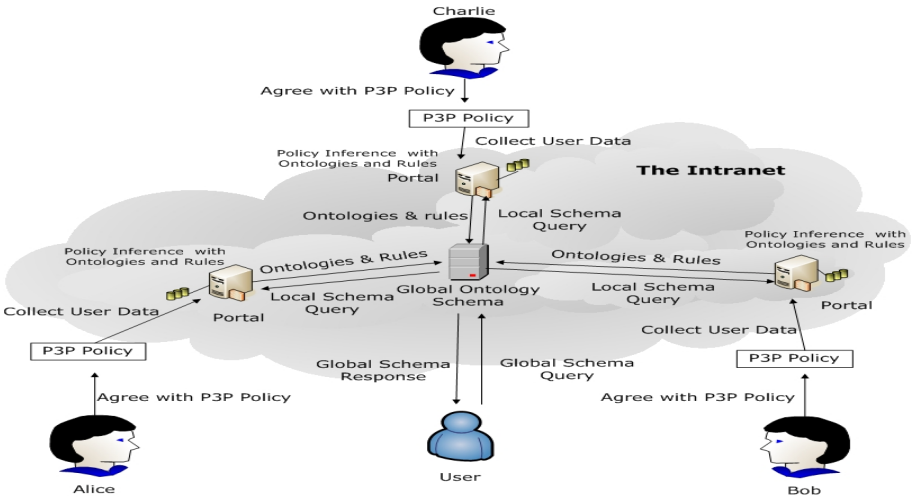

Fig. 1. A semantic privacy protection model extended from the P3P and EPAL integration for data integration and protection in a super-peer domain

Each enterprise as a peer declares its P3P privacy protection policies that takes the FIPs' criteria (see Figure 1). Then EPAL policies are established in each site, corresponding to the P3P [Karjoth and Schunter, 2002]. For each data request, the data handling and usage controls are based on the EPAL policies. However, P3P and EPAL lack formal and unambiguous semantics to specify privacy protection policies so they are limited in the policy enforcement and auditing support for software agents. One of the research challenges for the online privacy protection problem is to develop a privacy management framework and a formal semantics language to empower agents to enforce privacy protection policies. Agents must avoid any policy violation of each data request. We attempt to establish a semantic privacy protection model for a super-peer domain to address this issue. In a super-peer domain, each peer shares its collected data with other peers but without breaking the original data usage commitment to its clients [Karjoth et al., 2003].

[a]See http://www.privacyrights.org/ar/fairinfo.htm

1.1. *Research Issues and Contributions*

In this paper we are addressing the following *research issues*:

- We aim at providing data integration and protection services for various data sources to perform effective data sharing for different purposes in a super-peer domain,. The incentives for using a super-peer model involve the avoidance of solving the complex pair-wise ontology matching and rule integration problems between peers. In addition, various complex ontology evolution and compatibility issues among peers can be hidden in a super-peer domain.
- Privacy protection policy representation and enforcement issues are also addressed. Policies are expressed as a combination of ontologies and rules, i.e. $\mathtt{O} + \mathtt{R}$, where ontology `O` includes `TBox` schemas and `ABox` instances, and rule `R` includes deductive rule sets (`RS`) and facts (`F`). Data integration and protection are achieved at the super-peer for multiple peers through a combination of semantics-enabled formal protection policies (`FPP`).
- In a super-peer domain, the challenge of designing a semantic privacy protection model is to ensure a *soundness* and a *completeness* of data integration and protection within a super-peer domain. For the *soundness* criterion, we do not allow unintended data being released to the data users through the global policy schema (`GPS`) at the super-peer. Otherwise, it violates the privacy protection policies. As for the *completeness* criterion, we do not miss any eligible shared data when a user asks for a data request service at the super-peer. Therefore, shareable data obtained at the super-peer should equal data obtained directly from each peer.
- In the multiple super-peer domains environment, we focus on using an emergent semantic mapping technique from a super-peer domain to interconnect with another one when additional information is requested on demand. This wide-scale data integration and protection problem faces the challenge of effectiveness data sharing without causing any semantic ambiguity of ontology mappings among super-peers. In addition, we avoid the undecidable computation of query answering posed to the super-peer by using *acyclic* schema mappings in a tree-based information query.

Our contributions. Our main contributions are: (i) We offer a three layer semantic privacy-preserving model for a super-peer domain. This extends our previous work on data integration for privacy protection policies [Hu and Yang, 2011]. We define a formal policy using ontology for privacy protection concepts and rules for data query and access control services. (ii) We focus on solving the soundness and completeness of query rewriting problems for a super-peer domain by using a perfect ontology merging and rule integration. Followed by each possible data query at the super-peer, we briefly demonstrate how the soundness and completeness criteria of a privacy protection data integration can be achieved. (iii)In the multiple super-

peer domains environment, we propose a tree-based information query technique by using the GLAV semantic ontology mappings among super-peers to achieve a wide-scale data integration. This avoids possible cyclic schema mappings as shown in [Calvanese et al., 2006]. We also adopt the top-down query answering strategy to pose authorized view-based queries over the super-peer to provide data integration and protection services. By incrementally collecting global information from each additional super-peer domain, we use the GLAV schema mappings among super-peers to collect information from their peers by using the LAV mappings between a super-peer and its peers.

Outline. The paper is organized as follows. In Section 2, we present a semantic privacy-preserving model as a framework for data integration services. In Section 3, we define a formal policy combination as an integration of formal policies from autonomous data sources. Each formal policy is composed of ontologies and rules for each independent data source. A privacy protection policy is a type of formal policy used for specifying a data usage constraint from a data owner. In Section 4, we formally define a formal policy combination in terms of ontology mapping, alignment, and merging. Then, in Section 5, we demonstrate how a perfect rule integration is used for query rewriting at the super-peer corresponding to its local peers' schema. Following Section 6, the semantics of a super-peer data integration system is specified and demonstrated with an example. In Section 7, we briefly prove the soundness and completeness of privacy-preserving data integration for a super-peer domain. Finally, we point out the related work and draw our conclusion.

2. A Privacy-Preserving Model

A semantic privacy protection model is proposed with three layers for a super-peer domain, where the bottom layer provides data sources from the relational databases and the middle layer provides a semantics-enabled local schema for each peer's independent service domain. The top layer is served at the super-peer, which provides a unified global view of privacy-preserving data integration services (see Figure 2).

We have a merged global ontology schema created by mapping and aligning local ontology schemas with a LAV source description from multiple local schemas in the middle layer. The idea of using description logic (DL) to model the local and global schemas is to empower the ontology's abstract concept representation and reasoning capabilities. A query is defined as an SQWRL datalog rule in the SWRL-based policy to access a global ontology [O'Connor and Das, 2009]. Each SQWRL data service query posed to a global ontology at the super-peer is mapped to multiple queries as SQWRL datalog rules for local schemas. This is a LAV query rewriting service which has been investigated in databases but has largely been unexplored in the context of DL-based ontologies [Friedman et al., 1999].

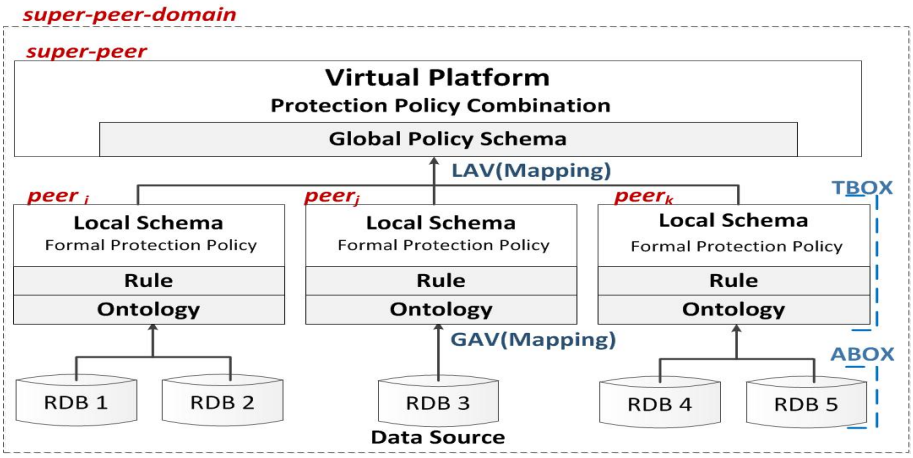

Fig. 2. A semantic privacy protection model in a super-peer domain

2.1. *Formal Privacy Protection Policy*

A policy's explicit representation in terms of ontologies or rules depends on what the underlying logic foundation of your policy language is. If your policies are created from a DL-based policy language, such as Rein or KAoS, then ordinary policies are shown as TBox schemas and ABox instances. Otherwise, policies are created from an LP-based policy language, such as EPAL or Protune. In that case, ordinary policies are a set of rules with predicates of unary, binary, or ternary variables and facts [Bonatti and Olmedilla, 2005].

A formal policy (FP) is a declarative expression corresponding to a human legal norm that can be executed in a computer system without causing any semantic ambiguity. An FP is created from a policy language (PL), and this PL is shown as a combination of ontology language and rule language. Therefore, an FP is composed of ontologies O and rules R, where ontologies are created from an ontology language and rules are created from a rule language.

A formal protection policy (FPP) is an FP that aims at representing and enforcing resource protection principles, where the structure of resources is modeled as ontologies O but the resources protection is shown as rules R.

A privacy protection policy shown as an FPP is a combination of ontologies and rules, e.g., O + R, where DL-based ontologies, such as OWL-DL ontologies provide a well-defined structure data model for data integration, while Logic Program (LP)-based rules, such as datalog rules provide further expressive power for data query and protection. There are numerous O + R combinations available for designing privacy protection policies, such as SWRL [Horrocks et al., 2005], and OWL2 RL [Grau et al., 2008b]. Each O + R combination implies whatever expressive power we can extract from ontologies for the rules and vice versa.

The SWRL is one of the O + R semantic web languages suitable for a policy

representation in the privacy protection model. However, this is not an exclusive selection. Other $\mathtt{O}+\mathtt{R}$ combinations, such as CARIN, OWL2 RL are also possible for modeling formal privacy protection policy whenever their underlying theoretical foundations and development tools are available. We fully utilize the SWRLTab development tools and SQWRL OWL-DL query language [O'Connor and Das, 2009] in the Protégé to model and enforce semantic privacy protection policies.

We face a research challenge of combining SWRL-based privacy protection policies from multiple peers to ensure the soundness and completeness of data integration and protection criteria in a super-peer domain. Another challenge is to solve the policy's syntax and semantics incompatibility when we allow policy combination in multiple peers. SWRL is based on the classical first order logic (FOL) semantics that mitigates a possible semantic and syntax inconsistency when policies come from different peers.

However, we still face a background policy inconsistency problem when default policy assumptions vary between different peers. For example, one peer uses open policy assumption, where no explicit option-out for data usage means option-in, but the other peer uses closed policy assumption, where no explicit option-in for data usage means option-out. We avoid this kind of policy inconsistency by requesting all sites to use a uniform policy assumption, and to only collect option-in data usage choices from users whenever multiple policies are integrated.

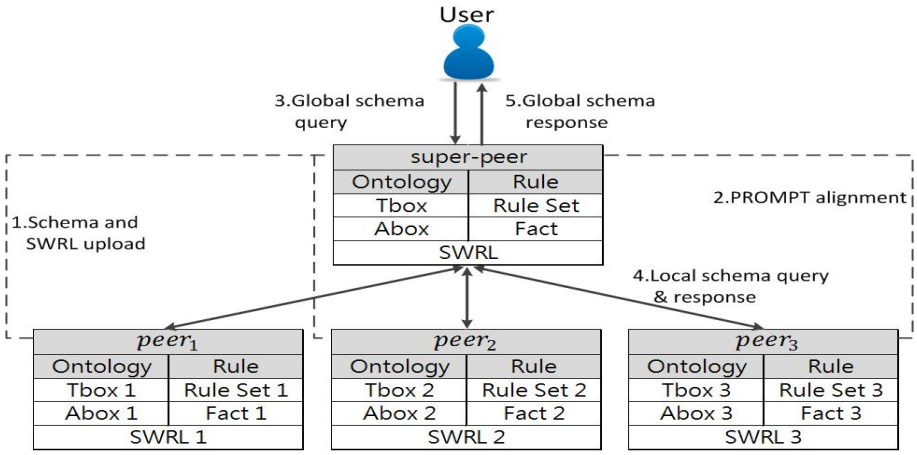

Fig. 3. A user quests information to the super-peer at a VP through ontology mapping, merging, and rule integration in a super-peer domain

Previous studies for policy combination did not consider solving the problem of merging multiple schemas and integrating access control rules from multiple peers [Bonatti et al., 2002] [Mazzoleni et al., 2008]. In this paper we propose a semantic privacy protection model that allows flexibly combining TBox's schemas of

privacy protection policies without moving ABox's instances from its original data source until a data request service is initiated (see Figure 3). Therefore, the global ontology's schemas and rules created at the super-peer have the latest updated incoming data from each peer when a user asks a query.

Data integration aims at providing unified and transparent access to a set of autonomous and heterogeneous data sources. The semantic privacy protection model providing global ontology schema for data integration is similar to the data integration problem solved by $\mathtt{DL-Lite_A}$ ontologies shown in [Calvanese et al., 2008a]. Here, we are also focusing on data protection besides data integration.

The goal of ontology-based data integration in $\mathtt{DL-Lite_A}$ is to provide a uniform access mechanism to a set of heterogeneous relational database sources, freeing the user from having the knowledge about where the data are, what are stored, and how they can be accessed. The idea is based on decoupling information access from its relational data storage so users only access the conceptual layer shown as ontology, while the relational data layer, hidden to users, manages the data.

Compared with $\mathtt{DL-Lite_A}$, we have extended and used it as a part of our semantic privacy protection model. We have three layers of data integration infrastructure instead of two layers shown in $\mathtt{DL-Lite_A}$ so we face a research challenge of ontology merging and rule integration from the middle layer to the top layer when we enforce a privacy protection policy (see Figure 3).

A semantic privacy protection model is composed of three main components:

- In the top layer, we have a global policy schema ($\mathtt{GPS_{super-peer}}$), including a global ontology schema ($\mathtt{GS_{super-peer}}$) aligned and merged from several local schemas ($\mathtt{LS_{peer}}$), e.g. TBox and a set of rule integration at the middle layer. The super-peer at this layer provides conceptual data access and protection services that give users a unified conceptual "global view" with access control power for each data request.
- Ontology-based data sources are external, independent, and heterogeneous, and each local ontology was combined with logic program (LP)-based rules for each peer in the middle layer.
- Mapping language (ML), semantically links a $\mathtt{GS_{super-peer}}$ and integrated rule set in the top layer to each peer's ontology $\mathtt{LS_{peer}}$ and privacy protection rules in the middle layer.

3. A Formal Policy Combination

A formal policy combination (FPC) in a global policy schema ($\mathtt{GPS_{super-peer}}$) allows data sharing as integration of FP from a variety of peers.

Each FP is shown as $\mathtt{K = O + R}$, where ontology $\mathtt{O} = (\mathtt{T}, \mathtt{A})$ and rule $\mathtt{R} = (\mathtt{RS}, \mathtt{F})$, T is TBox, and A is ABox; RS is a set of rules, and F is a set of facts.

$$\texttt{FPC} = \underset{i}{\oplus}\texttt{K}_i = (\underset{i}{\diamond}\texttt{O}_i, \underset{i}{\odot}\texttt{R}_i) = (\underset{i}{\diamond}(\texttt{T}, \texttt{A})_i, \underset{i}{\odot}(\texttt{RS}, \texttt{F})_i) = ((\underset{i}{\diamond}\texttt{T}_i, \underset{i}{\diamond}\texttt{A}_i), (\underset{i}{\odot}\texttt{RS}_i, \underset{i}{\odot}\texttt{F}_i))$$

where
i is the index of a peer i.
$\oplus$ is an operator for formal policy combination,
$\diamond$ is an operator for ontology mapping and merging,
$\odot$ is an operator for rule integration.

In a semantic privacy protection model, a formal protection policy combination (FPPC) allows data integration and protection from $\texttt{FPC} = \underset{i}{\oplus}\texttt{K}_i = (\underset{i}{\diamond}\texttt{O}_i, \underset{i}{\odot}\texttt{R}_i)$, where $\underset{i}{\odot}\texttt{R}_i = (\underset{i}{\odot}\texttt{RS}_i, \underset{i}{\odot}\texttt{F}_i)$ provides data query and protection services in $\underset{i}{\diamond}\texttt{O}_i$.

3.1. FPP *for Privacy Protection*

A privacy protection policy is a type of FPP. We designed an ontology that declares the FIPs' attributes as classes in an FPP (see Figure 4). The attributes, purpose, datauser, data, obligation, and action, allow people to specify the constraints of privacy protection policies using related property chains.

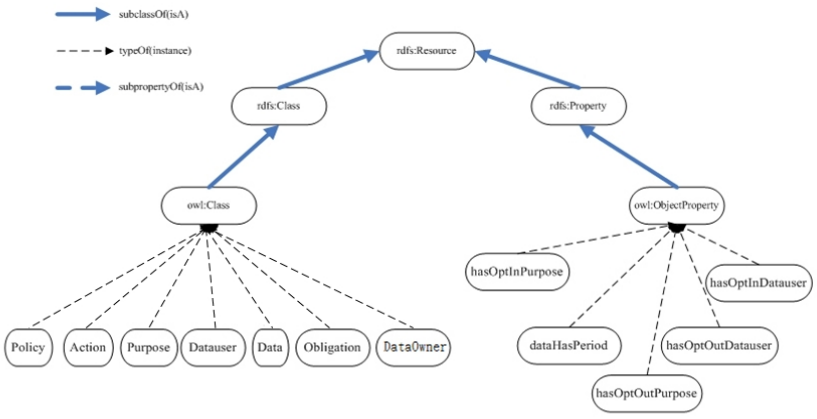

Fig. 4. A partial ontology schema for OECD FIPs' attributes shown as owl : Class, and constraints shown as owl : Property

Constraint properties is a type of owl : ObjectProperty that specifies the feasible domain and range classes of the above attributes. For example, a property of hasOptInPurpose has its domain and range classes shown as follows:

$\texttt{T} \sqsubseteq \forall\ \texttt{hasOptInPurpose}.\texttt{Data}$, $\texttt{T} \sqsubseteq \forall\ \texttt{hasOptInPurpose}^{-}.\texttt{Purpose}$.

Then a datalog rule, in the SWRL-based policy representation, allows us to use a property chain to combine the two feasible classes together:

$$\texttt{hasOptInPurpose.Data}(?data) \wedge \texttt{hasOptInPurpose}^{-}\texttt{.Purpose}(?purpose) \longrightarrow \texttt{hasOptInPurpose}(?data, ?purpose) \longleftarrow (1)$$

Similarly, a `hasOptInDatauser` property has its domain and range classes shown as follows:

$$\texttt{T} \sqsubseteq \forall\ \texttt{hasOptInDatauser.Data},\ \texttt{T} \sqsubseteq \forall\ \texttt{hasOptInDatauser}^{-}\texttt{.Datauser}.$$

Then, another datalog rule allows us to use another property chain to combine another two feasible classes together:

$$\texttt{hasOptInDatauser.Data}(?data) \wedge \texttt{hasOptInDatauser}^{-}\texttt{.Datauser}(?datauser) \longrightarrow \texttt{hasOptInDatauser}(?data, ?datauser) \longleftarrow (2)$$

Based on (1) and (2), we have a feasible set of `ABox` instances with `data`, `purpose`, and `datauser` combinations of an attribute set that was permitted from the original `dataowner` to allow a particular type of `datauser` to ask for a `data` set with a permissive `purpose`. When a peer collects a customer's data, the promise of data usage will be ensured if a data user's identity and usage purpose are verified successfully. Otherwise, the data will be kept secret without a data user's awareness.

These are easily extended to the other two attributes, `action` and `obligation`, to complete the FIPs' privacy protection criteria. An ordinary data user is allowed to ask a query service with `action` = `read` at the super-peer. The other actions, such as `deletion` or `modify`, are only allowed for a system administrator in the middle layer when (s)he asks to delete a user's data to satisfy the obligation of a data retention period or for a data owner to update his or her own profile data.

3.2. *Data Request Services*

A peer declares its privacy policy in P3P before a data owner's data is collected. Once a user accepts a peer's privacy declaration policy, the data usage constraints are specified as Figure 5, where FIP's five attributes $(?d, ?p, ?du, ?a, ?o)$ for `data`, `purpose`, `datauser`, `action`, and `obligation`, are classes, and `hasOptInDatauser`, `hasOptInPurpose`, etc., are properties proposed as chains of usage constraints for attributes. For each data request service, an initial feasible parameter input set is $\texttt{FS} = input(?du, ?r, ?p)$, where $?du \in$ `Datauser`, $?r = read \in$ `Action`, $?p \in$ `Purpose` and output dataset with associated obligations is $output(?d, ?o)$, where $?d \in$ `Data`, $?o \in$ `Obligation`. The feasible dataset shown as `ABox` instances will be discovered by using SQWRL datalog rules. Further permissible actions will be activated when the following data protection policies are

satisfied.

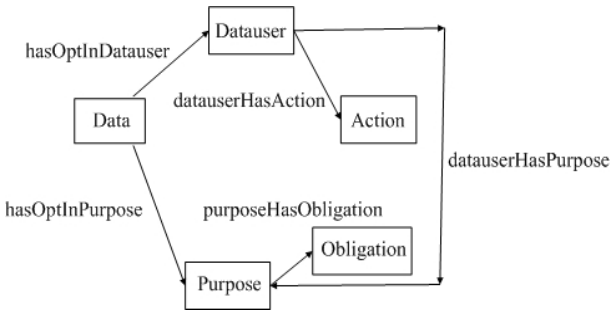

Fig. 5. Five major FIP's attributes, such as data, purpose, etc are shown as `owl` : `class` and chained by associated `owl` : `Property`, such as `hasOptInDatauser`, `hasOptInPurpose`, etc.

3.3. FPPC *at the super-peer*

A data user still possibly collects a shareable data by asking each peer individually without using a formal privacy protection policy combination (FPPC). However, the high complexity of using query services for all data sources hinders people from using this data integration approach. The other possible approach to collect a shareable data is to combine pair-wise peers' policies. Then, we face another scalability problem when more than two peers are intending to share their data.

In this semantic privacy-preserving model, we propose the super-peer infrastructure that allows a peer in each data source to offer its FPP at the super-peer to enforce FPPC. FPP in each data source is shown as $\mathtt{K} = \mathtt{O} + \mathtt{R}$, where ontology $\mathtt{O} = (\mathtt{T}, \mathtt{A})$ and rule $\mathtt{R} = (\mathtt{RS}, \mathtt{F})$. At the super-peer, we only map and merge T, e.g. TBox but leave A, e.g. ABox instances in its original RDB data source. Similarly, we only integrate RS, a set of rules at the super-peer but leave F, a set of facts in its original RDB data source. The benefit of using this approach is to map and merge the TBoxes and to integrate the RS with the updated data only once.

4. Ontology Mapping and Merging

A merged ontology comes from mapping and alignment that provides data integration services [Euzenat and Shvaiko, 2007]. In particular, data integration through ontologies, such as LAV is possible for multiple peers if a mapping language ML provides a semantic mapping description between the $\mathtt{GS}_{\mathtt{super-peer}}$ and the underlying $\mathtt{LS}_{\mathtt{peer}}$ of each peer [Friedman et al., 1999]. In LAV, the relationships between the $\mathtt{GS}_{\mathtt{super-peer}}$ and the $\mathtt{LS}_{\mathtt{peer}}$ are established by making LAV assertions. Every assertion has the form $Q_{LS_{peer}} \rightsquigarrow CQ_{GS_{super-peer}}$, where each vocabulary in the $Q_{LS_{peer}}$, i.e., `class` or `property` in a peer's local ontology schema, is defined as the views of

a conjunctive query (CQ) over the global schema $\mathtt{GS}_{\mathtt{super-peer}}$, so $CQ_{GS_{super-peer}}$ is a CQ over the global schema $\mathtt{GS}_{\mathtt{super-peer}}$ at the super-peer.

A $Q_{LS_{peer}} \rightsquigarrow CQ_{GS_{super-peer}}$ is defined as a privacy-aware authorized view of each peer so we do not disclose any non-shareable data to the super-peer whenever each peer submits its FPP for ontology merging and rule integration. A CQ can be defined as a subset of Datalog program, i.e. CQ containment problem, for querying the relational database. This problem was previously investigated in [Ullman, 2000].

On the other hand, the connection between the problem of answering queries using extensions of views and the problem rewriting queries using views were studied previously through an ontology expressed in DL [Goasdoué and Rousset, 2004]. In [Calvanese et al., 2008a], a relational data integration was obtained by mapping each ontology element, e.g. class and property, in the $\mathtt{LS}_{\mathtt{peer}}$ into an SQL query of a relational data source. This is a GAV approach that focuses on mapping the elements of the $\mathtt{LS}_{\mathtt{peer}}$ to a view (SQL query) over the sources.

4.1. *Perfect Ontology Alignment*

A mapping can be shown as (uid, e_1, e_2, n, ρ), where uid is a unique identity for the mapping, e_1, e_2 are entity names, such as class or property, and in the vocabulary of $\mathtt{O}_1$, $\mathtt{O}_2$, n is a numeric confidence measure between 0 and 1, and ρ is a relation such as subsumption ($\sqsubseteq$), equivalence ($\equiv$), or disjointness ($\perp$) between e_1 and e_2 [Jiménez-Ruiz et al., 2009].

In this study, the entity names for describing the ontology's class and property, and the structure of using these entity names in the root of the ontology schema for $\mathtt{O}_i$ to define the FIPs' privacy protection criteria (see Figure 5) are required to be the same. This is a strict constraint to achieve a perfect ontology alignment of this study. Moreover, a perfect mapping language ML provides semantic mappings for each entity $e \in \mathtt{GS}_{\mathtt{super-peer}}$ at the super-peer to the corresponding entities $e_i \in \mathtt{LS}_{\mathtt{peer}_i}$.

A perfect ontology alignment obtained via a mapping (uid, e_i, e_j, n, ρ) and merging between $\mathtt{T}_i$, i.e. TBox, in $\mathtt{O}_i$ and $\mathtt{T}_j$ in $\mathtt{O}_j$ satisfies the following conditions:

- $e_i \in \mathtt{T}_i$ and $e_j \in \mathtt{T}_j$ entity names are either defined for describing the root class names which correspond to the privacy protection concepts, such as `purpose`, `action`, `datauser`, `data`, and `obligation` or for property names, such as `hasOptInDatauser`, `hasOptInPurpose`, etc; Furthermore, entity names below the root class and root property are also defined for the descriptions of the underlying subclass and subproperty names.
- A numeric confidence measure n is always equal to 1.
- ρ is either equivalence ($\equiv$) or subsumption ($\sqsubseteq$) between entity names of $\mathtt{T}_i$ and $\mathtt{T}_j$ schemas. In an equivalent ($\equiv$) case, we can find a pair of one-to-one corresponding entity names for $e_i \in \mathtt{T}_i$ and $e_j \in \mathtt{T}_j$ in the same layer of the respective ontology schema with $n = 1$; In a subsumption ($\sqsubseteq$) case, there are subclass or subproperty entity names not in the same layer so $e_i \in \mathtt{T}_i$

and $e_i \sqsubseteq e_j \in \mathtt{T}_j$, and vice versa.

4.2. *Query Rewriting Services*

SWRL combines OWL-DL's ontology language with an additional datalog rule language, where a datalog rule language is shown as an axiom of ontology, a little extension of the OWL-DL language that overcomes the limitations of property chaining in the OWL-DL language [Horrocks et al., 2005]. The computation complexity of answering SWRL-based policies might be undecidable regarding the verification of rights access permission unless these policies satisfy the $\mathtt{DL-Safe}$ conditions [Motik et al., 2004].

SPARQL is a query language for the RDF(S)-based ontologies. OWL2 QL is another query language for the OWL2-based ontologies. We did not use SPARQL query language or OWL2 QL, since our current local and global ontologies are modeled as the OWL-DL ontology language. In fact, SPARQL might not be able to query the complete semantics of the OWL-DL's ontologies. The OWL-DL's ontology queries can be shown as the SQWRL datalog rules, where the CQ conditions are shown as the rule's body and the query results, i.e., views are shown as the rule's conclusion. SQWRL uses SWRL's strong FOL semantic foundation as its formal semantics so this query language provides a small but powerful array of operators that allows users to construct queries over OWL-DL ontologies [O'Connor and Das, 2009].

For each data request query service, a perfect mapping language ML provides the semantically linking of an entity name $e \in \mathtt{GS}_{\mathtt{super-peer}}$ in the datalog rule at the super-peer to the entity name $e_i \in \mathtt{LS}_{\mathtt{peer}_i}$ in the datalog rule at $\mathtt{peer_i}$, where $\mathtt{LS}_{\mathtt{peer}_i}$ is the TBox of $\mathtt{O}_i$, and e is a class or a property name. If there does not exist an $e_i \in \mathtt{TBox}_i$ in a subtree of the $\mathtt{LS}_{\mathtt{peer}_i}$ on the same layer as $e \in \mathtt{TBox}$ in the global tree of $\mathtt{GS}_{\mathtt{super-peer}}$, then we can recursively find a superclass or superproperty of e'_i with $e \sqsubseteq e'_i$ as the corresponding entity name, with a confidence measure of $n = 1$.

To successfully fulfill the semantically linking of any entity name $e \in \mathtt{GS}_{\mathtt{super-peer}}$ via ML, an ontology schema designer must follow the principles we propose using the specifications of concepts and relations for the FIPs on the root layer of each ontology's local schema's $\mathtt{LS}_{\mathtt{peer}_i}$. But we still allow the designer to use a different entity name string, $e_i \in \mathtt{LS}_{\mathtt{peer}_i}$ below the root layer of each local schema and to have an entirely different underlying subtree structure. We use *Prompt* ontology mapping algorithm [Euzenat and Shvaiko, 2007] first to synchronize the entity names between $\mathtt{LS}_{\mathtt{peer}_i}$ and further perform the ontology mappings and aligning operations. Finally, we perfectly merge their schemas even if the subtrees of the local schemas are variant.

We use ML to map the name of a class entity $c \in \mathtt{GS}_{\mathtt{super-peer}}$ to one of the equivalent local ontology schema's class entity names in a deeper subtree, say $c_j \in \mathtt{LS}_{\mathtt{peer}_j}$, i.e., $c \leftrightsquigarrow c_j$ in the datalog rule's conditions of each data request service. When the class semantics for c is $c \sqsubseteq c_i$ in the $\mathtt{LS}_{\mathtt{peer}_i}$, i.e., we do not have

a corresponding class $c_i' \in \mathtt{LS}_{\mathtt{peer}_i}$ on the same lower layer of a schema tree as $c \in \mathtt{GS}_{\mathtt{super-peer}}$. All of the `ABox` instances a_i in the class name entity c_i, i.e., $a_i \in c_i$ are still feasibly collected for this data request. This is because class c_i is a legal domain class or range class for a particular property in the datalog rule for enforcing its privacy protection.

Similarly, a property $p \in \mathtt{GS}_{\mathtt{super-peer}}$ is mapped to another equivalent property $p_j \in \mathtt{LS}_{\mathtt{peer}_j}$ for the associated datalog rule's body conditions. Then property $p \leftrightsquigarrow p_j$ might be on a lower layer in the schema tree when compared with property $p_i \in \mathtt{LS}_{\mathtt{peer}_i}$. We still regard property p_i as feasible for its enforcement of the datalog rule on data integration and protection. Finally, if we consider mappings for binding property and class from the aligning ontology schema $\mathtt{GS}_{\mathtt{super-peer}}$ to $\mathtt{LS}_{\mathtt{peer}_i}$ and $\mathtt{LS}_{\mathtt{peer}_j}$ to the respective datalog rule, then we have the following semantically linking relationships by using `ML`'s mapping to align the ontology's class and property shown as follows:

Property $\mathtt{p} \in \mathtt{GS}_{\mathtt{super-peer}}$ with its domain class `dc` and range class `rc` that are mapped to property $\mathtt{p}_i \in \mathtt{LS}_{\mathtt{peer}_i}$ with its domain class $\mathtt{dc}_i$ and its range class $\mathtt{rc}_i$. For each data request service using a perfect mapping language `ML`, when $\mathtt{p} \sqsubseteq \mathtt{p}_i$, we use property $\mathtt{p}_i$. Otherwise, when $\mathtt{p}_i \sqsubseteq \mathtt{p}$, we use property $\mathtt{p}$ for the datalog rule $\mathtt{r}_i$. When $\mathtt{dc} \sqsubseteq \mathtt{dc}_i$ and $\mathtt{rc} \sqsubseteq \mathtt{rc}_i$, we use class $\mathtt{dc}_i$ and $\mathtt{rc}_i$. Otherwise, when $\mathtt{dc}_i \sqsubseteq \mathtt{dc}$ and $\mathtt{rc}_i \sqsubseteq \mathtt{rc}$, we use class `dc` and `rc` for the datalog rule $\mathtt{r}_i$.

Here, we did not explicitly consider an algebra operations, such as intersection or union, for `class/subclass` with property as shown in OWL-DL. Intuitively, this class/subclass and property algebra operation problem can be transformed to the generic class/property problem when terms from different data sources can be mapped and aligned at the super-peer.

Example 4.1. *In Figure 6, after we map and align two local partial ontology schemas,* $\mathtt{LS}_{\mathtt{peer}_i}$ *and* $\mathtt{LS}_{\mathtt{peer}_j}$*, into a merged partial ontology global schema* $\mathtt{GS}_{\mathtt{super-peer}}$*, we receive a data request service with class* $\mathtt{P}_{212}$*. In the purpose class* $\mathtt{P}$*,* $\mathtt{P}_{111} \leftrightsquigarrow \mathtt{P}_{211}$*, but* $\mathtt{P}_{212} \in \mathtt{GS}_{\mathtt{super-peer}}$ *does not have a corresponding subclass in* $\mathtt{LS}_{\mathtt{peer}_i}$*, since* $\mathtt{P}_{212} \sqsubseteq \mathtt{P}_{21}$ *and* $\mathtt{P}_{21} \leftrightsquigarrow \mathtt{P}_{11}$*. When a data request service asks for class* $\mathtt{P}_{212} \in \mathtt{GS}_{\mathtt{super-peer}}$*, mapping language* `ML` *will map* $\mathtt{P}_{212}$ *to* $\mathtt{P}_{11}$ *for the datalog rule* $\mathtt{r}_i$ *to query the* $\mathtt{LS}_{\mathtt{peer}_i}$*.*

5. Perfect Rule Integration

In `FPPC`, we define an integrated rule set $\underset{i}{\odot}\mathtt{R}_i = (\underset{i}{\odot}\mathtt{RS}_i, \underset{i}{\odot}\mathtt{F}_i)$ to enforce data query and protection services in $\underset{i}{\diamond}\mathtt{O}_i$. In fact, an integrated rule set $\underset{i}{\odot}\mathtt{RS}_i$ is a part of `FPC` that was created by collecting the datalog rules, e.g. SQWRL queries, in formal policies $\mathtt{FP}_i$, from local peers. A datalog rule $\mathtt{r}_i$ in the $\mathtt{R}_i$ of $\mathtt{FP}_i$ is shown as:

$$\mathtt{H} \longleftarrow \mathtt{B}_1 \wedge \mathtt{B}_2 \wedge, \cdots, \wedge \mathtt{B}_n,$$

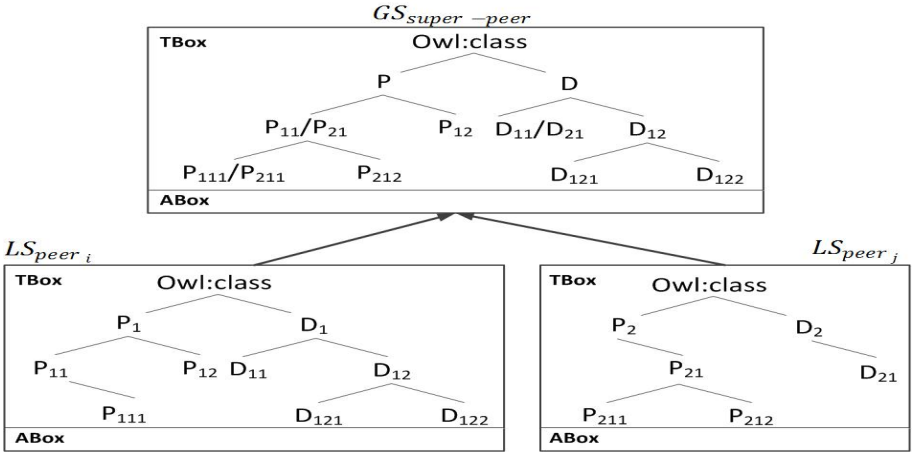

Fig. 6. A partial ontology mapping for class alignment and ontology merging

where $\mathtt{H}$, the query results (or views) are expressed as SQWRL built-ins, such as $\mathtt{sqwrl:select}$ and the rule antecedent $\mathtt{B}_i$, are defined as a pattern matching specifications, i.e., query conditions that are either SQWRL built-ins or class and property predicates from the ontology schema. More specifically, this datalog rule is related to a $\mathtt{CQ}$ of the form:

$$\mathtt{H}(\overrightarrow{x}) \leftarrow \exists \overrightarrow{y}(\mathtt{B}_1(\overrightarrow{x}, \overrightarrow{y}) \wedge \mathtt{B}_2(\overrightarrow{x}, \overrightarrow{y}) \wedge, \cdots, \wedge \mathtt{B}_n(\overrightarrow{x}, \overrightarrow{y}))$$

where $\mathtt{B}_i(\overrightarrow{x}, \overrightarrow{y})$ is a conjunction of atoms with $\mathtt{L}_A$, the set of function-free first-order logic formulas, involving the free variables (also the *distinguished* variables of the query) $\overrightarrow{x} = x_1, \cdots, x_n$, and the existentially quantified variables (also the *non-distinguished* variables of the query), $\overrightarrow{y} = y_1, \cdots, y_n$. $\mathtt{H}(\overrightarrow{x})$ is the views of query results posed over the super-peer to perform data integration within a super-peer domain.

A perfect rule integration is defined for the integration of any datalog rules as: $\exists \mathtt{r}_i \in \mathtt{RS}_i$ in $\mathtt{FP}_i$, for the purpose of data integration and protection without causing conflicts with $\exists \mathtt{r'}_i \in \underset{i}{\odot}\mathtt{R}_i$, $\lambda_i \in \underset{i}{\diamond}\mathtt{O}_i$, i.e., conditions do not exist for $\exists \mathtt{r}_i \models \lambda_i \Rightarrow \exists \mathtt{r'}_i \not\models \lambda_i$, or $\exists \mathtt{r}_i \not\models \lambda_i \Rightarrow \exists \mathtt{r'}_i \models \lambda_i$. Then, $\exists \mathtt{r'}_i \in \underset{i}{\odot}\mathtt{R}_i$ at the super-peer can be activated and mapped by the perfect mapping language $\mathtt{ML}$ into $\mathtt{r}_i$, posed to a super-peer to enable a global data query and protection service of multiple peers within a super-peer domain.

Example 5.1. *A rule* $\mathtt{r'}_i$ *is one of the rules within the integrated rule set at the super-peer. When a user asks for a data set* ?$\mathtt{d}$ *with related obligations* ?$\mathtt{o}$ *under the feasible parameter input set* $\mathtt{FS}_i = (\mathtt{M1}, \mathtt{TMarketing6}, \mathtt{Read2})$*, where data user* $\mathtt{M1}$ *is a marketing staff with the purpose of achieving telephone mark-*

ing TMarketing*, A rule* $\mathtt{r'}_i$ *is mapped to a rule* $\mathtt{r}_i$ *and a rule* $\mathtt{r}_j$ *using the rule mapping processes when we have done an upward perfect ontology mapping, alignment, merging and a perfect rule integration. A perfect mapping language* ML *with downward operation maps the* $\mathtt{r'}_i$*'s predicates, such as class, property to the corresponding predicates in a rule* $\mathtt{r}_i$ *and a rule* $\mathtt{r}_j$ *with* $\mathtt{MUser(M1)} \sqsubseteq \mathtt{Datauser(M1)}$, $\mathtt{TMarketing(TMarketing6)} \sqsubseteq \mathtt{Purpose(TMarketing6)}$*. Therefore, real data query and protection services requested by a rule* $\mathtt{r'}_i$ *are performed by a rule* $\mathtt{r}_i$ *and a rule* $\mathtt{r}_j$*.*

A rule $\mathtt{r'}_i$ *query posed to the super-peer at the* $\underset{i}{\diamond}\mathtt{O}_i$*:*

$\underline{\mathtt{MUser(M1)}} \wedge \underline{\mathtt{TMarketing(TMarketing6)}}$
$\wedge \mathtt{datauserHasPurpose(M1, TMarketing6)}$
$\wedge \mathtt{datauserHasAction(M1, Read2)}$
$\wedge\ \mathtt{hasOptInPurpose(?d, TMarketing6)}$
$\wedge \mathtt{hasOptInDataUser(?d, M1)}$
$\wedge\ \mathtt{purposeHasObligation(TMarketing6, ?o)}$
$\longrightarrow \mathtt{sqwrl} : \mathtt{selectDistinct(?d, M1, TMarketing6, Read2, ?o)}$

A rule $\mathtt{r}_i$ *query posed to a* $\mathtt{peer_i}$ *at the* $\mathtt{O}_i$*:*

$\underline{View(\mathtt{Datauser(M1)})} \wedge \underline{View(\mathtt{TMarketing(TMarketing6)})}$
$\wedge\ \mathtt{datauserHasPurpose(M1, TMarketing6)}$
$\wedge\ \mathtt{datauserHasAction(M1, Read2)}$
$\wedge\ \mathtt{hasOptInPurpose(?d, TMarketing6)}$
$\wedge\ \mathtt{hasOptInDataUser(?d, M1)}$
$\wedge\ \mathtt{purposeHasObligation(TMarketing6, ?o)}$
$\longrightarrow \mathtt{sqwrl} : \mathtt{selectDistinct(?d, M1, TMarketing6, Read2, ?o)}$

A rule $\mathtt{r}_j$ *query posed to a* $\mathtt{peer_j}$ *at the* $\mathtt{O}_j$*:*

$\underline{View(\mathtt{MUser(M1)})} \wedge \underline{View(\mathtt{Purpose(TMarketing6)})}$
$\wedge\ \mathtt{datauserHasPurpose(M1, TMarketing6)}$
$\wedge\ \mathtt{datauserHasAction(M1, Read2)}$
$\wedge\ \mathtt{hasOptInPurpose(?d, TMarketing6)}$
$\wedge\ \mathtt{hasOptInDataUser(?d, M1)}$
$\wedge\ \mathtt{purposeHasObligation(TMarketing6, ?o)}$
$\longrightarrow \mathtt{sqwrl} : \mathtt{selectDistinct(?d, M1, TMarketing6, Read2, ?o)}$

6. Semantics of a Super-Peer Data Integration System

Inspired by [Calvanese et al., 2006] [Halevy et al., 2004], we define a super-peer data integration system as a set of super-peer domains $\Pi = \{\pi_1, \pi_2, ..., \pi_n\}$, where each super-peer domain π_j is an autonomous information site that exports its information content in terms of the super-peer $\mathtt{sp}_j$'s schema to another super-peer domain.

6.1. *Semantics of a Super-Peer Domain*

In each super-peer domain π_i, actual data is stored in a set of local data sources $\mathtt{DS} = \{\mathtt{ds_1}, \mathtt{ds_2}, ..., \mathtt{ds_m}\}$. Using the GAV local mappings, we associate a set of local peer $\mathtt{P} = \{\mathtt{peer_1}, \mathtt{peer_2}, ..., \mathtt{peer_n}\}$ in π_i with each individual ontology schema to the views of the related relational data sources, i.e., SQL queries. Furthermore, through LAV semantic mappings, a set of peers P's local ontology schemas are also mapped and aligned into the super-peer sp's global view. Formally, the semantics of a super-peer domain is based on SWRL, a subset of the classical first-order logic (FOL) semantics, that mitigates a possible semantic and syntax inconsistency when data sources come from different peers. This FOL semantics approach is different from the multi-modal epistemic logic approach used in [Calvanese et al., 2006]. We avoid any possible cyclic schema mappings among a large number of peers by using a tree-based information query through schema mappings among the super-peers. This cyclic avoidance technique is similar to the Piazza PDMS approach [Halevy et al., 2004]. This technique not only simplifies the unrestricted mapping among multiple peers but it also meets the recent development trend of using a description logic (DL) technique, such as $\mathtt{DL-Lite_A}$, for data integration [Calvanese et al., 2008a] [Calvanese et al., 2008b].

A super-peer domain $\pi \in \Pi$ is defined as a tuple (P, SPD, GS, LS, M, DS):

- A super-peer sp is the only node in a super-peer domain $\pi \in \mathtt{SPD}$, which allows an agent to enforce the global protection policies. This enforcement action empowers the super-peer sp to facilitate information collection through a $\mathtt{CQ}_\pi(\mathtt{sp})$ posed to the $\mathtt{GS_{sp}}$ in the super-peer sp of π.
- Through the local LAV mapping assertions, a global schema $\mathtt{GS_{super-peer}}$ provides an integrated view for a set of peers from P in a π. Similar to the Section 4 technique, we proposed that every LAV assertions has the form $Q_{LS_{peer}} \rightsquigarrow CQ_{GS_{super-peer}}$, where $Q_{LS_{peer}}$ provides the views of the CQ over the global schema, $\mathtt{GS_{super-peer}}$ for each peer. $CQ_{GS_{super-peer}}$ is a CQ over the global schema $\mathtt{GS_{super-peer}}$ at the super-peer.
- A set of peers from P are mediators. A peer $\mathtt{p} \in \pi$ maps its local ontology schema $\mathtt{LS_{peer}}$ to a set of relational data sources, $\mathtt{ds_i}$, from DS in π. Therefore, this query uses the unfolding GAV mapping assertions. $Q_{LS_{peer}} \rightsquigarrow CQ_{ds_i}$, where $Q_{LS_{peer}}$ is a vocabulary of an ontology local schema of a peer that maps to the SQL CQ over a set of data sources, $\mathtt{ds_i}$, from DS.
- A set of local mapping assertions, M, created from a mapping language ML, are used to semantically link between a super-peer sp and a set of peers from P in a π. The semantics of a set of global mapping assertions among super-peers will be addressed in Section 8.
- A set of local data sources, $\mathtt{ds_i}$ from DS, are relational structure data that store the materialized instances.

7. Soundness and Completeness

In this section, we briefly demonstrate how the exact query rewriting service satisfies the soundness and completeness criteria by using the LAV source descriptions based on the $\mathtt{GPS_{sp}} = (\underset{i}{\diamond}\mathtt{O}_i, \underset{i}{\odot}\mathtt{R}_i)$: If $\mathtt{CQ_{sp}}$ is a conjunctive query over $\underset{i}{\diamond}\mathtt{O}_i$ at the super-peer $\mathtt{sp}$ within a super-peer domain π, and $\mathtt{CQ_{peer_i}}$ is a conjunctive query over $\mathtt{O}_i$ using LAV source descriptions from $\mathtt{peer_i}$, then $\forall x \quad \mathtt{CQ_{sp}}(x) \longleftrightarrow \bigsqcup_i \mathtt{CQ_{peer_i}}(x)$. In [Goasdoué and Rousset, 2004], authors showed that when a query has a finite number of *maximally contained conjunctive rewritings*, the complete set of its answers can be obtained as the union of the answer sets of its rewritings. The *datalog-rewriting* was introduced, in which query language is a *hybrid language* with CARIN as its combination of $\mathtt{O} + \mathtt{R}$, and the rewriting language is a *relational language*. They also provided a rewriting algorithm, and showed that the *RewriteQuery* is sound and complete.

In comparison, we use LAV for rewriting queries and use SWRL as a combination of $\mathtt{O} + \mathtt{R}$. A perfect ontology merging and a rule integration ensure the soundness and completeness of data integration in the semantic privacy-preserving model. This will be briefly shown as follows:

7.1. *[Soundness]*

For the *soundness* criterion, we do not allow any unintentionally released (or protected) data for a user by using a query rewriting service with a rule (query) $\mathtt{r'}_i \in \underset{i}{\odot}\mathtt{R}_i$ at the super-peer $\mathtt{sp}$ within a super-peer domain π.

Theorem 7.1. ***[Soundness]*** *After a perfect ontology alignment and rule integration with* FPPC, $\exists\mathtt{GPS_{sp}} = (\underset{i}{\diamond}\mathtt{O}_i, \underset{i}{\odot}\mathtt{R}_i)$ *at the super-peer* $\mathtt{sp}$ *within a super-peer domain* π*, Under a particular feasible parameter input set* $\mathtt{FS}_i$*, if* $\lambda_j \in \mathtt{O}_i$ *is protected by a* $\mathtt{FPP}_i$ *at each* $\mathtt{peer_i}$*, i.e.,* $\forall i, r_i \in \mathtt{R}_i \nvDash \lambda_j$*, then* $\mathtt{r'}_i \in \underset{i}{\odot}\mathtt{R}_i \nvDash \lambda_j$ *for the same* $\mathtt{FS}_i$*, where* λ_j *is a protective data set in* $\mathtt{O}_i$*.*

Proof. (Sketch) If $\mathtt{CQ_{sp}}$ is a conjunctive query over $\underset{i}{\diamond}\mathtt{O}_i$ at the super-peer $\mathtt{sp}$ within a super-peer domain π and $\mathtt{CQ_{peer_i}}$ is a conjunctive query over $\mathtt{O}_i$ in a $\mathtt{peer_i}$, then we need to prove the statement $\forall x \;\; CQ_{sp}(x) \longrightarrow \bigsqcup_i CQ_{peer_i}(x)$. This statement is equivalent to the original argument: If $r_i \in \mathtt{R}_i \nvDash \lambda_j$, then $\mathtt{r'}_i \in \underset{i}{\odot}\mathtt{R}_i \nvDash \lambda_j$. The $\mathtt{CQ_{sp}}(x)$ is a query containment of datalog rule $\mathtt{r'}_i$ and the $\mathtt{CQ_{peer_i}}(\mathtt{x})$ is a query containment of datalog rule $r_i \in \mathtt{R}_i$. The statement $\forall x \;\; CQ_{sp}(x) \longrightarrow \bigsqcup_i CQ_{peer_i}(x)$ is true because the LAV schema mapping only allows the protected concept λ_j in each $\mathtt{peer_i}$ to be connected to the global schema. After using a perfect ontology alignment and a perfect rule integration with a perfect mapping language ML, we avoid the following condition: $\exists \mathtt{r}_i \nvDash \lambda_j \Rightarrow \exists \mathtt{r'}_i \models \lambda_j$. □

7.2. *[Completeness]*

As for the *completeness* criterion, we do not allow any eligible shared data being missed for a query by a query rewriting service with a rule (query) $\mathtt{r'}_i \in \underset{i}{\odot}\mathtt{R}_i$ at the super-peer $\mathtt{sp}$ within a super-peer domain π.

Theorem 7.2. ***[Completeness]*** *After a perfect ontology alignment and rule integration with* FPPC, $\exists\mathtt{GPS}_{\mathtt{sp}} = (\underset{i}{\diamond}\mathtt{O}_i, \underset{i}{\odot}\mathtt{R}_i)$ *at the super-peer* $\mathtt{sp}$ *within a super-peer domain* π*, Under a particular feasible parameter input set* $\mathtt{FS}_i$*, if* $\lambda_j \in \mathtt{O}_i$ *is shareable by a* $\mathtt{FPP}_i$ *at each* $\mathtt{peer_i}$*, i.e.,* $\forall i, r_i \in \mathtt{R}_i \models \lambda_j$*, then* $\mathtt{r'}_i \in \underset{i}{\odot}\mathtt{R}_i \models \lambda_j$ *for the same* $\mathtt{FS}_i$*, where* λ_j *is a shareable data set in* $\mathtt{O}_i$*.*

Proof. (Sketch) If $\mathtt{CQ}_{\mathtt{sp}}$ is a conjunctive query over $\underset{i}{\diamond}\mathtt{O}_i$ at the super-peer $\mathtt{sp}$ within a super-peer domain π and $\mathtt{CQ}_{\mathtt{peer_i}}$ is a conjunctive query over $\mathtt{O}_i$ in a $\mathtt{peer_i}$, then we need to prove the statement $\forall x \;\; CQ_{sp}(x) \longleftarrow \bigsqcup_i CQ_{peer_i}(x)$. This statement is equivalent to the original argument: If $r_i \in \mathtt{R}_i \models \lambda_j$, then $\mathtt{r'}_i \in \underset{i}{\odot}\mathtt{R}_i \models \lambda_j$. The $\mathtt{CQ}_{\mathtt{sp}}(x)$ is a query containment of datalog rule $\mathtt{r'}_i$ and the $\mathtt{CQ}_{\mathtt{peer_i}}(\mathtt{x})$ is a query containment of datalog rule $r_i \in \mathtt{R}_i$. The statement $\forall x \;\; CQ_{sp}(x) \longleftarrow \bigsqcup_i CQ_{peer_i}(x)$ is true because the LAV schema mapping only allows the protected concept λ_j in each $\mathtt{peer_i}$ to be connected to the global schema. After using a perfect ontology alignment and a perfect rule integration with a perfect mapping language ML, we avoid the following condition: $\exists \mathtt{r}_i \models \lambda_j \Rightarrow \exists \mathtt{r'}_i \not\models \lambda_j$. □

8. Semantics of Multiple Super-Peer Domains

A super-peer domain π_i is related to other super-peer domains π_j by means of a set of super-peer GLAV semantic mapping assertions for *tree-based* information query. A super-peer's semantic mapping is shown as follows:

$$CQ_{\pi_j}(sp_j) \rightsquigarrow CQ_{\pi_i}(sp_i)$$

where $CQ_{\pi_j}(sp_j)$ is a conjunctive query over the super-peer $\mathtt{sp_j}$ in a super domain $\pi_j \in \Pi$, and $CQ_{\pi_i}(sp_i)$ is a conjunctive query over the super-peer $\mathtt{sp_i}$ in a super domain $\pi_i \in \Pi$. A $CQ_{\pi_j}(sp_j)$ is defined as a privacy-aware authorized view of a super-peer domain π_j whenever the super-peer $\mathtt{sp}_j$ intends to export its shareable information in terms of its schema $\mathtt{GS}_{\mathtt{sp_j}}$ mapping to another super-peer domain π_i's $\mathtt{sp_i}$ schema $\mathtt{GS}_{\mathtt{sp_i}}$ through the super-peers' GLAV semantic mapping assertions. Note that in the super-peer system Π, only the tree-based information query with the GLAV schema mappings are imposed on the topology of super-peer mapping assertions, hence the graph corresponding to Π is acyclic. The tree-based datalog rule corresponding to $\pi_j \in \Pi$ contains mapping from one super-peer $\mathtt{sp_j}$'s shareable

ontology schema symbol, $\mathtt{R_j}$ to another super-peer $\mathtt{sp_i}$'s ontology schema symbol, $\mathtt{R_i}$. Therefore one edge from the super-peer $\mathtt{sp_j}$'s symbol, $\mathtt{R_j}$, to the super-peer $\mathtt{sp_i}$'s symbols, $\mathtt{R_i}$, exists if there is a super-peer mapping assertion in Π whose tail mentions $\mathtt{R_j}$ and whose head mentions $\mathtt{R_i}$ in a datalog rule.

Example 8.1. *Under the data protection law, Hospitals* A, B, *and* C, *in three super-peer domains are allowed to share their patients' Electronic Health Records (*EHRs*) after patients give their consents for medication (see Figure 7). A patient,* Jong, *was hospitalized in Hospital* A *for surgery. After that,* Jong *went to Hospital* B *for an outpatient medication. A physician, Matt, in Hospital* C *was authorized by* Jong *to query his shareable* EHRs *collected from Hospitals* A *and* B*'s super-peers, sp_a, sp_b for a medical treatment. The partial ontology global schemas for Hospital* A, B, *and* C *are: GS_{sp_a}, GS_{sp_b}, and GS_{sp_c} (see Figure 8).*

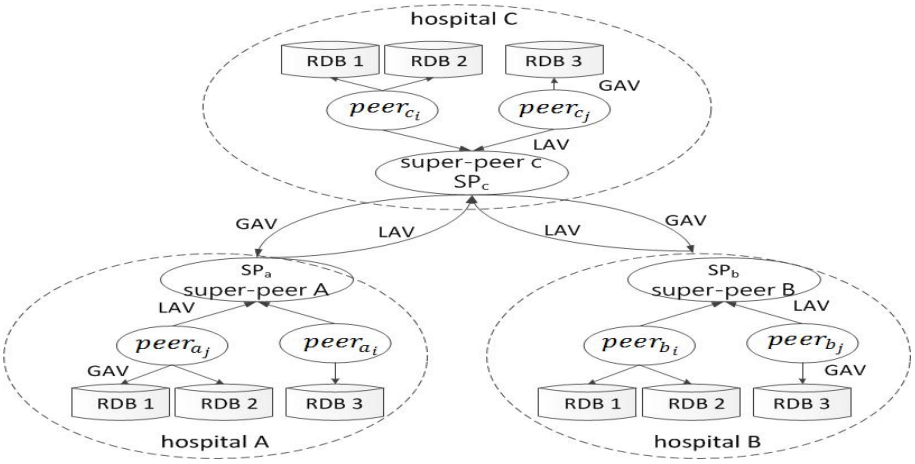

Fig. 7. The tree-based information query technique to share the hospitals A, B, C's EHRs through the GAV and LAV (GLAV) semantic mappings among GS_{sp_a}, GS_{sp_b}, and GS_{sp_c} in their super-peers

Hospital A *has the following terms as its super-peer's ontology global schema, GS_{sp_a}, vocabularies:*

Class*:* Clinic, HealthData, SurgeryData, *and* HospitalizationData
Property*:* create *and* beTreated *with the respective domain and range class:*

$\mathtt{T} \sqsubseteq \forall\ \mathtt{create.Clinic}$, $\mathtt{T} \sqsubseteq \forall\ \mathtt{create}^{-}\mathtt{.HealthData}$
$\mathtt{T} \sqsubseteq \forall\ \mathtt{beTreated.Individual}$, $\mathtt{T} \sqsubseteq \forall\ \mathtt{beTreated}^{-}\mathtt{.Clinic}$.

Hospital B *has the following terms as its super-peer's ontology global schema, GS_{sp_b}, vocabularies:*

Class*:* Person*,* HealthCenter*, and* PatientData *with subClass* OutPatientData
Property*:* own*,* beMedicated *with their respective domain and range class are:*

$\mathtt{T} \sqsubseteq \forall\ \mathtt{own.Person}$, $\mathtt{T} \sqsubseteq \forall\ \mathtt{own}^{-}\mathtt{.PatientData}$.
$\mathtt{T} \sqsubseteq \forall\ \mathtt{beMedicated.Person}$, $\mathtt{T} \sqsubseteq \forall\ \mathtt{beMedicated}^{-}\mathtt{.HealthCenter}$.

Hospital C *has the following terms as its super-peer's ontology global schema, GS_{sp_c}, vocabularies:*

Class*:* Patient*,* Hospital*,* Surgery*, and* HealthRecord
Property*:* beCured*,* hasHealthRecord*,* generate*, and* hasMedType *with their respective domain and range class are:*

$\mathtt{T} \sqsubseteq \forall\ \mathtt{beCured.Patient}$, $\mathtt{T} \sqsubseteq \forall\ \mathtt{beCured}^{-}\mathtt{.Hospital}$
$\mathtt{T} \sqsubseteq \forall\ \mathtt{hasHealthRecord.Patient}$, $\mathtt{T} \sqsubseteq \forall\ \mathtt{hasHealthRecord}^{-}\mathtt{.HealthRecord}$
$\mathtt{T} \sqsubseteq \forall\ \mathtt{generate.Hospital}$, $\mathtt{T} \sqsubseteq \forall\ \mathtt{generate}^{-}\mathtt{.HealthRecord}$
$\mathtt{T} \sqsubseteq \forall\ \mathtt{hasMedType.HealthRecord}$, $\mathtt{T} \sqsubseteq \forall\ \mathtt{hasMedType}^{-}\mathtt{.Outpatient}$
$\mathtt{T} \sqsubseteq \forall\ \mathtt{hasMedType}^{-}\mathtt{.Surgery}$, $\mathtt{T} \sqsubseteq \forall\ \mathtt{hasMedType}^{-}\mathtt{.Hospitalization}$

Use LAV approach to define each class and property of Hospitals A *and* B *super-peers, sp_a and sp_b's, global schemas as views in terms of Hospital* C*'s sp_c's global schema vocabularies are shown as follows:*

Views use at the sp_c in Hospital C *created from the* $\mathtt{GS}_{\mathtt{sp_a}}$ *schema's vocabularies are:*

$\mathtt{def}(\mathtt{V1}_{\mathtt{Individual}}) \overset{\mathtt{LAV}}{\subseteq} \mathtt{Patient}$,
$\mathtt{def}(\mathtt{V2}_{\mathtt{Clinic}}) \overset{\mathtt{LAV}}{\subseteq} \mathtt{Hospital}$,
$\mathtt{def}(\mathtt{V3}_{\mathtt{HealthData}}) \overset{\mathtt{LAV}}{\subseteq} \mathtt{HealthRecord}$
$\mathtt{def}(\mathtt{V4}_{\mathtt{SurgeryData}}) \overset{\mathtt{LAV}}{\subseteq} \mathtt{HealthRecord} \wedge \forall \mathtt{hasMedType.Surgery}$
$\mathtt{def}(\mathtt{V5}_{\mathtt{HospitalizationData}}) \overset{LAV}{\subseteq} \mathtt{HealthRecord} \wedge \forall \mathtt{hasMedType.Hospitalization}$
$\mathtt{def}(\mathtt{V6}_{\mathtt{create}}) \overset{\mathtt{LAV}}{\subseteq} \mathtt{generate}$,
$\mathtt{def}(\mathtt{V7}_{\mathtt{hold}}) \overset{\mathtt{LAV}}{\subseteq} \mathtt{hasHealthRecrod}$,
$\mathtt{def}(\mathtt{V8}_{\mathtt{beTreated}}) \overset{\mathtt{LAV}}{\subseteq} \mathtt{beCured}$,
$\mathtt{def}(\mathtt{V9}_{\mathtt{Purpose}}) \overset{\mathtt{LAV}}{\subseteq} \mathtt{Purpose}$

Views use at the sp_c in Hospital C *created from the* $\mathtt{GS}_{\mathtt{sp_b}}$ *schema's vocabularies are:*

$\mathtt{def}(\mathtt{V10}_{\mathtt{Person}}) \overset{\mathtt{LAV}}{\subseteq} \mathtt{Patient}$,

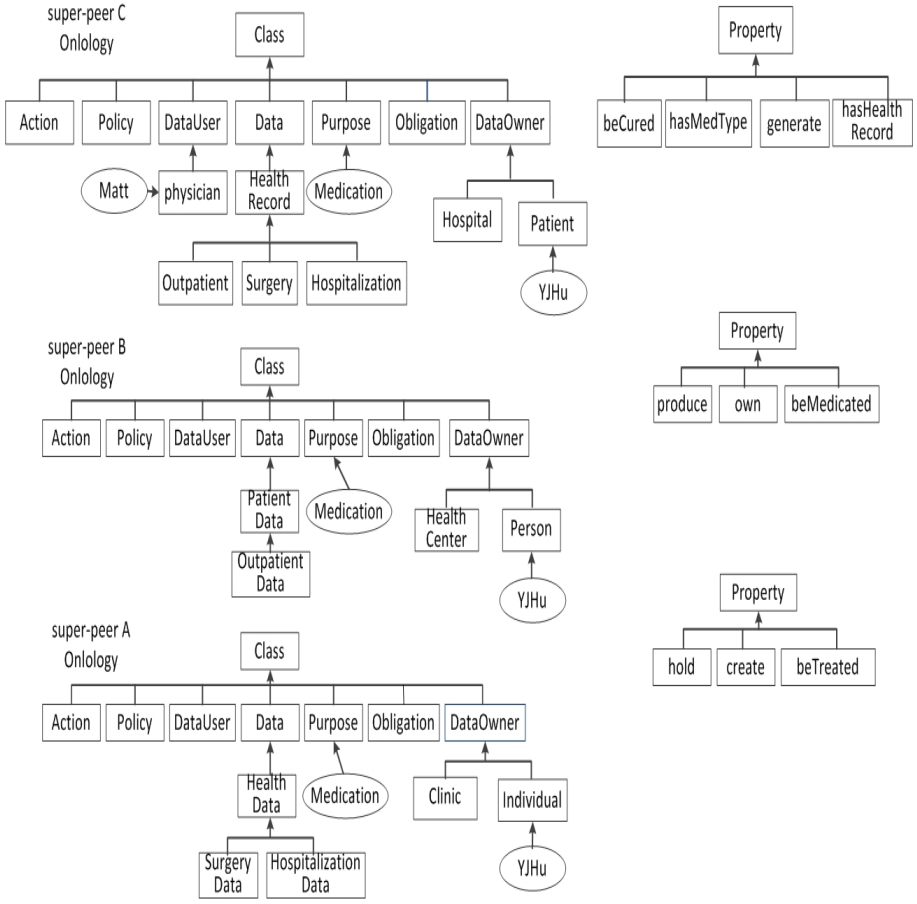

Fig. 8. A partial ontology for EHRs' sharing and privacy protection

$\texttt{def}(\texttt{V11}_{\texttt{HealthCenter}}) \overset{\texttt{LAV}}{\subseteq} \texttt{Hospital}$

$\texttt{def}(\texttt{V12}_{\texttt{PatientData}}) \overset{\texttt{LAV}}{\subseteq} \texttt{HealthRecord},$

$\texttt{def}(\texttt{V13}_{\texttt{OutPatientData}}) \overset{\texttt{LAV}}{\subseteq} \texttt{HealthRecord} \wedge \forall\ \texttt{hasMedType.OutPatient}$

$\texttt{def}(\texttt{V14}_{\texttt{produce}}) \overset{\texttt{LAV}}{\subseteq} \texttt{generate},$

$\texttt{def}(\texttt{V15}_{\texttt{own}}) \overset{\texttt{LAV}}{\subseteq} \texttt{hasHealthRecrod},$

$\texttt{def}(\texttt{V16}_{\texttt{beMedicated}}) \overset{\texttt{LAV}}{\subseteq} \texttt{beCured},$

$\texttt{def}(\texttt{V17}_{\texttt{produce}}) \overset{\texttt{LAV}}{\subseteq} \texttt{generate},$

$\texttt{def}(\texttt{V18}_{\texttt{Purpose}}) \overset{\texttt{LAV}}{\subseteq} \texttt{Purpose}$

A physician, Matt, queries a patient's EHRs *at the* sp_c *of Hospital* C *by using a query*

rewriting service instead of directly requesting each hospital. An original datalog-based rule for a conjunctive query $CQ_{\pi_c}(sp_c)$ at Hospital C *is shown as:*

$\mathtt{sp_c} : \mathtt{Patient}(\mathtt{YJHu}) \wedge \mathtt{sp_c} : \mathtt{beCured}(\mathtt{YJHu}, ?\mathtt{y}) \wedge \mathtt{sp_c} : \mathtt{hasHealthRecrod}(\mathtt{YJHu}, ?\mathtt{r})$
$\wedge\ \mathtt{sp_c} : \mathtt{HealthRecord}(?\mathtt{r}) \wedge \mathtt{sp_c} : \mathtt{hasMedType}(?\mathtt{r}, ?\mathtt{mt}) \wedge$
$\mathtt{sp_c} : \mathtt{generate}(?\mathtt{y}, ?\mathtt{r}) \wedge \mathtt{sp_c} : \mathtt{Purpose}(\mathtt{Medication})$
$\longrightarrow \mathtt{sp_c} : \mathtt{HealthRecord}(\mathtt{YJHu}, ?\mathtt{r})$

$Q_{\pi_c}(sp_c : HealthRecord(YJHu, ?r)) \overset{GAV}{\longleftarrow}$
$CQ_{\pi_a}(sp_a : HealthData(YJHu, ?r)) \wedge CQ_{\pi_b}(sp_b : PatientData(YJHu, ?r))$

Query rewriting of the $CQ_{\pi_c}(sp_c)$ in terms of two CQ*s, e.g., $CQ_{\pi_a}(sp_a)$ and $CQ_{\pi_b}(sp_a)$, uses views defined at the $Q_{\pi_c}(sp_c)$:*

$\mathtt{V1}_{\mathtt{Individual}} \wedge \mathtt{V8}_{\mathtt{beTreated}} \wedge \mathtt{V7}_{\mathtt{hold}} \wedge \mathtt{V4}_{\mathtt{SurgeryData}} \wedge \mathtt{V6}_{\mathtt{create}} \wedge \mathtt{V9}_{\mathtt{Purpose}}$
$\longrightarrow \mathtt{sp_a} : \mathtt{HealthData}(\mathtt{YJHu}, ?\mathtt{r}) \leftrightsquigarrow CQ_{\pi_a}(sp_a)$

Above $CQ_{\pi_a}(sp_a)$ is corresponding to:

$\mathtt{sp_a} : \mathtt{Individual}(\mathtt{YJHu}) \wedge \mathtt{sp_a} : \mathtt{beTreated}(\mathtt{YJHu}, ?\mathtt{c}) \wedge \mathtt{sp_a} : \mathtt{hold}(\mathtt{YJHu}, ?\mathtt{d})$
$\wedge\ \mathtt{sp_a} : \mathtt{SurgeryData}(?\mathtt{sd}) \wedge \mathtt{sp_a} : \mathtt{create}(?\mathtt{h}, ?\mathtt{hd}) \wedge \mathtt{sp_a} : \mathtt{Purpose}(\mathtt{Medication})$
$\longrightarrow \mathtt{sp_a} : \mathtt{HealthData}(\mathtt{YJHu}, ?\mathtt{sd})$

$\mathtt{V10}_{\mathtt{Person}} \wedge \mathtt{V16}_{\mathtt{beMedicated}} \wedge \mathtt{V15}_{\mathtt{own}} \wedge \mathtt{V13}_{\mathtt{OutPatientData}} \wedge \mathtt{V17}_{\mathtt{produce}} \wedge \mathtt{V18}_{\mathtt{Purpose}}$
$\longrightarrow \mathtt{sp_b} : \mathtt{PatientData}(\mathtt{YJHu}, ?\mathtt{r}) \leftrightsquigarrow CQ_{\pi_b}(sp_b)$

Above $CQ_{\pi_b}(sp_b)$ is corresponding to:

$\mathtt{sp_b} : \mathtt{Person}(\mathtt{YJHu}) \wedge \mathtt{sp_b} : \mathtt{beMedicated}(\mathtt{YJHu}, ?\mathtt{c}) \wedge \mathtt{sp_b} : \mathtt{own}(\mathtt{YJHu}, ?\mathtt{d})$
$\wedge\ \mathtt{sp_b} : \mathtt{OutPatientData}(?\mathtt{od}) \wedge \mathtt{sp_b} : \mathtt{produce}(?\mathtt{h}, ?\mathtt{hd})$
$\wedge\ \mathtt{sp_b} : \mathtt{Purpose}(\mathtt{Medication}) \longrightarrow \mathtt{sp_b} : \mathtt{PatientData}(\mathtt{YJHu}, ?\mathtt{od})$

9. Related Work

Data integration is a pervasive challenge faced in the applications that need to query across multiple autonomous and heterogeneous data sources. This problem has been receiving considerable attention from researchers in the fields of Artificial Intelligence and Database System more than a decade [Halevy et al., 2006] [Levy, 2001]. A logic of the Description Logic (DL) family is used to model the ontology managed by the integration system, to formulate queries posed to the system, and to perform several types of automated reasoning supporting both the modeling, and the query answering process [Calvanses et al., 2002]. The ontology expresses the

domain of interest of the information system at a high level of abstraction, and the relationship between data at the sources and instances of concepts and roles in the ontology is expressed by means of mappings, such as GLAV, GAV, LAV [Calvanese et al., 1998] [Poggi et al., 2008].

Recently, various studies, such as PAYGO systems, have pointed out using a PDMS in the wide-scale data integration system [Halevy et al., 2003] [Madhavan et al., 2007]. This inspires us to put forth efforts in the peer data integration research. However, PAYGO systems used a relation data model, which did not use any ontology-based conceptual data modeling for schema mappings and information query. This hampers the feasibility of information integration when the semantics for describing real world entities is represented as an abstract concept. In the peer-to-peer data integration systems [Calvanese et al., 2006], authors used multi-modal epistemic formalization to describe each peer as a rational agent that exchanges knowledge/belief with other peers in a two-level peer-to-peer architecture. This epistemic modeling is far more complex and infeasible in the real data integration system implementation. Therefore, the description logic $\mathtt{DL-Lite_A}$, a subset of the first-order logic (FOL), adopted from the semantic web technique was proposed to solve the traditional data integration problem [Calvanese et al., 2008a] [Calvanese et al., 2008b]. In this paper, we use a three-level super-peer architecture with tree-based information query through the GLAV schema mappings among super-peers to avoid any possible peer-to-peer cyclic mapping problems.

Data integration is usually hampered by legitimate and widespread privacy concerns, so it is critical to develop a technique to enable the integration of data that does not lose privacy. We face a challenge to develop a privacy framework for data integration that is flexible and clear to the end users [Clifton et al., 2004]. View-based query answering over DL provides a framework to answer a query under the assumption that the only accessible information consists of the precomputed answers to a set of queries, called views. Privacy-aware access to data, each user is associated with a set of views, called authorization views, which specify the information that the user is allowed to access [Calvanese et al., 2008b].

The EFAF access control model is an extension of the FAF that provided the solution for privacy protection [Jajodia et al., 2001] [Karjoth and Schunter, 2002]. This method is close to our solution, but its privacy protection control is more on the logic program and less on the ontology schema used for structure data modeling. This also prevents the data integration and protection in multiple sites. The other similar model for enforcing the enterprise privacy protection goes to the EPAL [Karjoth et al., 2003] [Vimercati et al., 2007]. Another OASIS XACML is a policy language for privacy and digital rights protection. However, it is an XML-based policy language so the policies based on XACML possibly might have ambiguous semantics that prevent us from using a flexible policy combination in multiple peers [Anderson, 2006].

10. Conclusion and Further Study

We propose a semantic privacy protection model which encompasses and extends the existing works on data sharing and integration through a super-peer data management. We intend to solve the privacy protection problem to provide data integration and integration in the multiple peers by using one of the ontology and rule language combinations, e.g. SWRL. Another OWL2 combination will be considered in the near future [Grau et al., 2008b]. In addition, this model can be extended to a modular reuse of ontologies for data integration and protection in the cross-domain cloud computing environment [Grau et al., 2008a] [Hu et al., 2011].

The perfect ontology alignment through ontology mapping and merging creates a global ontology schema at the super-peer by integrating multiple peers' local ontology schemas. In addition, the perfect rule integration by the perfect mapping language avoids any possible data usage conflicts between datalog rules from different data sources at the super-peer. In fact, a datalog rule is considered to be a conjunctive query, which provides data query and protection services for each peer.

However, this perfect ontology alignment is impossible without the restrictions of using the same ontology schema in the root layers of multiple peers. We face another policy hidden conflict challenge when background default policy assumptions vary between different peers. The semantics-enabled policies are combined at the super-peer, so we simplify the data integration and protection services for a PDMS.

In a wide-scale data integration scenario, we use the tree-based information query through the super-peers' GLAV semantic mapping assertions to avoid cyclic schema mappings among super-peers. Therefore, we can incrementally collect information from the other super-peer domains. The soundness and completeness criteria are preserved for data integration in a super-peer domain. This supports the trustworthiness of a policy combination for data integration and protection from multiple peers.

We currently did not deal with the issue of privacy-preserving instance fusion (or record linkage) in our tree-based information query as shown in [Hall and Fienberg, 2010] [Rahm et al., 2005]. All of these need further study.

Acknowledgements

This research was partially supported by the NSC Taiwan under Grant No. NSC 100-2221-E-004-011-MY2.

References

Anderson, A. H. (2006). A comparison of two privacy policy languages: EPAL and XACML. In *Proc. of the 3rd ACM Workshop on Secure Web Services (SWS'06)*, pages 53–60. ACM.

Antón, I. A. et al. (2007). A roadmap for comprehensive online for privacy policy management. *Comm. of the ACM*, 50(7):109–116.

Beneventano, D. et al. (2007). Querying a super-peer in a schema-based super-peer network. In Moro, G. et al., editors, *Databases, Information Systems, and Peer-to-Peer Computing*, LNSC, pages 13–25. Springer.

Bernstein, A. P. and Haas, L. M. (2008). Information integration in the enterprise. *Comm. of the ACM*, 51(8):72–79.

Bonatti, A. P. et al. (2002). An algebra for composing access control policies. *ACM Trans. on Information and Systems Security*, 5(1):1–35.

Bonatti, P. and Olmedilla, D. (2005). Policy language specification, enforcement, and integration. project deliverable D2, working group I2. Technical report, REWERSE.

Calvanese, D. et al. (1998). Description logic framework for information integration. In *Proc. of the 6th Int. Conf. on Principles of Knowledge Representation and Reasoning*, pages 2–13. Morgan Kaufmann.

Calvanese, D. et al. (2006). Data management in peer-to-peer data integration systems. *Global Data Management*, pages 177–201.

Calvanese, D. et al. (2008a). Data integration through $DL-Lite_A$ ontologies. In *3rd Int. Workshop on Semantics in Data and Knowledge Base (SDKB)*, volume 4925, pages 26–47. Springer.

Calvanese, D. et al. (2008b). View-based query answering over description logic ontologies. In *Proc. of KR-2008*. AAAI Press.

Calvanese, D. and Giacomo, D. G. (2005). Data integration: A logic-based perspective. *AI Magazine*, 26(1):59–70.

Calvanese, D. and toehrs (2004). Logical foundations of peer-to-peer data integration. In *Proc. of the 23rd ACM SIGACT SIGMOD SIGART Sym. on Principles of Database Systems PODS-2004*, pages 241–251.

Calvanses, D. et al. (2002). Description logics for information integration. In *Computational Logic*, LNAI 2408, pages 41–60. Springer.

Clifton, C. et al. (2004). Privacy-preserving data integration and sharing. In *Data Mining and Knowledge Discovery*, pages 19–26. ACM.

Euzenat, J. and Shvaiko, P. (2007). *Ontology Matching*. Springer-Verlag.

Franklin, M., Halevy, A., and Maier, D. (2005). From databases to dataspaces: A new abstraction for information management. *SIGMOD Record*, 34:27–33.

Friedman, M. et al. (1999). Navigational plans for data integration. In *Proc. of the Sixteen National Conference on Artificial Intelligence (AAAI'99)*, pages 67–73. AAAI/MIT Press.

Goasdoué, F. and Rousset, M. C. (2004). Answering queries using views: a KRDB perspective for the semantic web. *ACM Trans. on Internet Technology*, 4(3):255–288.

Grau, C. B. et al. (2008a). Modular reuse of ontologies: Theory and practice. *Journal of Artificial Intelligence Research*, pages 273–318.

Grau, C. B. et al. (2008b). OWL2: The next step for OWL. *Web Semantics: Science, Services and Agents on the World Wide Web 3*, pages 309–322.

Halevy, A. et al. (2003). Schema mediation in peer data management systems. In *Proc. 19th Int. Conference on Data Engineering (ICDE)*, pages 505–516.

Halevy, A. et al. (2004). The Piazza peer data management system. *Knowledge and Data Engineering, IEEE Transactions on*, 16(7):787 – 798.

Halevy, A., Rajaraman, A., and Ordille, J. (2006). Data integration: The teenage years. In *VLDB'06*, pages 9–16. ACM.

Halevy, Y. A. (2001). Answering queries using views: A survey. *The VLDB Journal*, 10(4):270–294.

Hall, R. and Fienberg, E. S. (2010). Privacy-preserving record linkage. In *Proc. of the 2010 Int. Conf. on Privacy in Statistical databases*. Springer.

Horrocks, I. et al. (2005). OWL rules: A proposal and prototype implementation. *Web Semantics: Science, Services and Agents on the World Wide Web 3*, 3(1):23–40.

Hu, Y. J., Wu, W. N., and Yang, J. J. (2011). Semantics-enabled policies for information sharing and protection in the cloud. In *Proc. of 3rd Int. Conf. on Social Semantics*, LNCS.

Hu, Y. J. and Yang, J. J. (2011). A semantic privacy-preserving model for data sharing and integration. In *International Conference on Web Intelligence, Mining and Semantics (WIMS'11)*. ACM Press.

Jajodia, S. et al. (2001). Flexible support for multiple access control policies. *ACM Trans. on Database Systems*, 26(2):214–260.

Jiménez-Ruiz, E. et al. (2009). Ontology integration using mappings: Towards getting the right logical consequences. In *ESWC 2009*, LNCS 5554, pages 173–187. Springer.

Karjoth, G. and Schunter, M. (2002). A privacy policy model for enterprises. In *15th IEEE Computer Security Foundations Workshop (CSFW)*. IEEE.

Karjoth, G., Schunter, M., and Herreweghen, v. E. (2003). Translating privacy practices into privacy promises - how to promise what you can keep. In *POLICY'03*. IEEE.

Lenzerini, M. (2002). Data integration: A theoretical perspective. In *Proceedings of the ACM Symposium on Principles of Database Systems (PODS)*, pages 233–246. ACM.

Levy, Y. A. (2001). Logic-based techniques in data integration. In Yu, T. and Jajodia, S., editors, *Logic-based Artificial Intelligence*, pages 1–27. Kulwer.

Madhavan, J. et al. (2007). Web-scale data integration: You can only afford to pay as you go. In *Proc. of CIDR-07*.

Mazzoleni, P. et al. (2008). XACML policy integration algorithms. *ACM Trans. on Information and System Security*, 11(1).

Motik, B., Sattler, U., and Studer, R. (2004). Query answering for OWL-DL with rules. In *3rd International Semantic Web Conference (ISWC) 2004*, LNCS 3298, pages 549–563. Springer.

Nash, A. and Deutsch, A. (2007). Privacy in GLAV information integration. In *ICDT 2007*, LNCS 4353, pages 89–103. Springer.

O'Connor, J. M. and Das, K. A. (2009). SQWRL: a query language for OWL. In *OWLED*, volume 529. CEUR.

Poggi, A. et al. (2008). Linking data to ontologies. *Journal on Data Semantics X*, 4900:133–173.

Rahm, E. et al. (2005). iFuice - information fusion utilizing instance correspondences and peer mappings. In *WebDB 2005*, pages 7–12.

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

Vimercati, S. D. C. d. et al. (2007). Access control policies and languages in open environments. In Yu, T. and Jajodia, S., editors, *Secure Data Management in Decentralized Systems*, pages 21–58. Springer.