A Semantic Privacy-Preserving Model for Data Sharing and Integration

Yuh-Jong Hu   
ENT Lab., Dept. of CS  
National Chengchi University  
Taipei, Taiwan, 11605  
u@cs.nccu.edu.tw

Jiun-Jan Yang   
ENT Lab., Dept. of CS  
National Chengchi University  
Taipei, Taiwan, 11605  
98753036@nccu.edu.tw

ABSTRACT
In this paper, we encompass and extend previous ontology-based data integration system. A semantic privacy-preserving model provides authorized view-based query answering over a widespread multiple servers for data sharing and integration. The combined semantics-enabled privacy protection policies are used to empower the data integration and access control services at the virtual platform (VP). The perfect rules integration of datalog rules enforces the data query and protection services. Semantics-enable policies are combined together at the VP, but the access control criteria specified in each server are still satisfied. Therefore the soundness and completeness of data sharing and protection criteria are ensured to support the validity of policy combination. This guarantees the trustworthiness of data sharing and protection services in multiple servers.

Categories and Subject Descriptors
H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—query formulation; H.3.5 [Information Storage and Retrieval]: Online Information Services—data sharing; K.4.1 [Computers and Society]: Public Policy Issues—privacy, regulation

General Terms
WWW, Semantic Web, Database

1. INTRODUCTION
Large enterprises spend a great deal of time and money on data (or information) integration [3]. Data integration is the problem of combining the data from autonomous and heterogeneous sources, and providing users with a unified view of these data through so called global (or mediated) schema. The design of a data integration system is a very complex task, which includes several different issues: heterogeneity of the data sources, relation between the global schema and the data sources, and how to process queries expressed on the global schema, etc [11].

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 [10] [26]. 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, allowing flexible schema definitions independent of the particular details of the data sources [14] [30]. The data integration system uses these different source descriptions to reformulate a user query into a query over the source schemas. However, data sharing and integration are hampered by legitimate and widespread privacy concerns so it is critical to develop techniques that enables the integration and sharing of data without losing a user’s privacy [12].
Privacy protection policies represent a long-term promise made by an enterprise to its users and are determined by business practice 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 enable 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 [2].

Although many organizations post online privacy policies, they must realize that simply posting a privacy policy on their websites does not guarantee true compliance with existing legislation. Following the OECD's Fair Information Principles (FIPs)\(^1\), an organization should provide norms of personal information process 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.

In this paper we are addressing the following research issues. More detailed modelling and implementation will be shown in the later sections.

- Data sharing and protection services are considered in a large number of servers. The incentives for using the virtual platform (VP) is to avoid solving the complex pair-wise problem of ontology matching and rule integration between these servers. Therefore a unified global data sharing and protection service can be achieved at the VP.

- Privacy protection policies are expressed as a combination ontology and rule, i.e. \(O + R\), where ontology \(O\) includes TBox schema and ABox instances, and rules \(R\) include deductive rule set (RS) and facts (F). Data sharing and protection in multiple servers are achieved through a combination of semantics-enabled formal protection policy (FP).\(^2\)

- The challenge of designing a semantic privacy protection model is to ensure a soundness and a completeness of data sharing and protection in multiple servers. For the soundness criterion, we do not allow unintended data being released to the data users through the global policy schema (GPS) at the VP. 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 VP. Therefore, shareable data obtained at the VP should equal data obtained directly from each server.

Each enterprise server declares its P3P privacy protection policies that takes into account the FIPs criteria (see Figure 1). Then EPAL policies are established in each site, corresponding to the P3P [24]. 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 the 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 to address this issue. Each server shares its collected data with other servers but without breaking the original data usage commitment to its clients [25].

The contributions of this paper are twofold. We first offer a three layers semantic privacy-preserving model which encompasses and extends the existing work on data sharing and integration by using a combination of ontology and rule for the representation of privacy protection policies. In particular, we define a formal policy using ontology for privacy protection concept descriptions and rule for data query and access control services. Then we focus on solving the soundness and completeness of query rewriting problem using a perfect ontology merging and a perfect rule integration from the local formal protection policies. Followed by each possible data query at the VP, we briefly demonstrate how the soundness and completeness criteria for privacy protection data integration can be achieved using this semantics-enabled privacy-preserving model.

The paper is organized as follows. In section 2, we present a semantic privacy-preserving model as a framework for data sharing and 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, merging, and alignment. Then we demonstrate how a perfect rule integration is used for query rewriting at the VP corresponding to each local schema. In section 6, we briefly prove the soundness and completeness of privacy-preserving data sharing and integration based on this semantic privacy-preserving model. We conclude with related work and discussion in the last two sections.

2. A PRIVACY-PRESERVING MODEL

A semantic privacy protection model is proposed with three layers, where the bottom layer provides data sources from the relational databases, the middle layer provides a semantics-enabled local schema for each independent service domain. The top layer is served at the VP, which provides a unified global view of privacy-preserving data sharing and 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
Figure 1: A semantic privacy protection model extended from the integration of P3P and EPAL for data sharing and protection in multiple servers

is defined as an SQWRL datalog rule in the SWRL-based policy to access to a global ontology [31]. Each SQWRL data service query for a global ontology at the VP is mapped to multiple queries as SQWRL datalog rules for each local schema. This is a LAV query rewriting service which has been investigated in databases but it is largely unexplored in the context of DL-based ontologies [14].

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 DL-based policy language, such as Rein or KAoS, then ordinary policies are shown as TBox schema and ABox instances. Otherwise, policies created from LP-based policy language, such as EPAL or Protune ordinary policies are a set of rules with predicates of unary, binary, or ternary variables and facts [5].

In the SemPIF framework [21], we define Policy Interchange Format (PIF) to follow W3C O+R standards [6] and strives to provide a mechanism for agents to preserve different policy syntax and semantics throughout its policy integration and interchange. In addition, agents can use meta-PIF, providing further management and reconciliation services of PIF-enabled multiple policies across various domains. In this paper, we apply the SemPIF framework for the privacy-preserving data integration through a combination of formal policies.

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 modelled 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 sharing, 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 [20], and OWL2 RL [17]. Each O + R combination implies what 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. But this is not an exclusive selection. Other O + 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 [31] 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 servers to ensure the soundness and completeness of data sharing and protection criteria. Another challenge is to solve the policy’s syntax and semantics incompatibility when we allow policy combination in multiple servers. SWRL is based on the classical first order logic (FOL) semantics that mitigates a possible semantic and syntax inconsistency when policies come from different servers.

But we still face a background policy inconsistency problem when default policy assumptions vary between different
servers. For example, one server uses open policy assumption, where no explicit option-out for data usage means option-in, but the other server 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 collect option-in data usage choices from users whenever multiple policies are integrated.

Previous studies for policy combination did not consider solving the problem of merging multiple schemas and integrating access control rules from multiple servers [4] [28]. In this paper we propose a semantic privacy protection model that allows flexibly combining TBoxes of privacy protection policies without moving ABox instances from its original data source until a data request service is initiated (see Figure 3). Therefor the global ontology TBox schema and rules created at the VP have the latest updated incoming data from each server 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 sharing is similar to the data integration problem solved by DL – LiteA ontologies shown in [8]. Here we are also focusing on data protection besides data sharing and integration.

The goal of ontology-based data integration in DL – LiteA 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 they 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 DL – LiteA, we have extended and used it as a part of our semantic privacy protection model. We have three layers of data sharing and integration infrastructure instead of two layers shown in DL – LiteA 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 composed of three main components:

- In the top layer at the VP, we have a global policy schema (GPS), including a global ontology schema (GS) aligned and merged from several local schemas (LS), e.g. TBox and a set of rule integration at the middle layer. The VP 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 server in the middle layer.

- Mapping language (ML), which semantically links a GS and integrated rule set in the top layer to each server’s ontology LS and privacy protection rules in the middle layer.

3. A FORMAL POLICY COMBINATION

A formal policy combination (FPC) in a global policy schema (GPS) allows data sharing as integration of FP from a variety of servers.

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

\[
\text{FPC} = \oplus K_i = (\otimes \mathcal{O}_i, \circ \mathcal{R}_i) = (\otimes (T_i, A_i), \circ (RS_i, F_i)) = ((\otimes T_i, \circ A_i), (\otimes RS_i, \circ F_i))
\]
where $i$ is the index of a server $i$, 
⊕ is an operator for formal policy combination, 
⋄ is an operator for ontology mapping and merging, 
⊙ is an operator for rule integration.

In a semantic privacy protection model, a formal protection policy combination ($FPPC$) allows data sharing and protection from 
$$FPP = \oplus K = (\odot O, \odot R),$$
where $\odot R = (\odot RS, \odot F)$ provides data query and protection services in $\odot O$.

### 3.1 Privacy Protection Policy (PFP)

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 that allow people to specify the constraints of privacy protection policies using related property chains.

Constraint properties is a type of $owl\text{:ObjectProperty}$ that specify what are the feasible domain and range classes of the above attributes. For example, a property $\text{hasOptInPurpose}$ has its domain and range classes shown as follows:

$$T \subseteq \forall \text{hasOptInPurpose}.\text{Data},$$

$$T \subseteq \forall \text{hasOptInPurpose}^\neg.\text{Purpose}.$$  

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

$$\text{hasOptInPurpose}.\text{Data}(?data) \wedge \text{hasOptInPurpose}^\neg.\text{Purpose}(?purpose) \rightarrow \text{hasOptInPurpose}(?data, ?purpose) \leftarrow (1)$$

Similarly, a $\text{hasOptInDatauser}$ property has its domain and range classes shown as follows:

$$T \subseteq \forall \text{hasOptInDatauser}.\text{Data},$$

$$T \subseteq \forall \text{hasOptInDatauser}^\neg.\text{Datauser}.$$  

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

$$\text{hasOptInDatauser}.\text{Data}(?data) \wedge \text{hasOptInDatauser}^\neg.\text{Datauser}(?datauser) \rightarrow \text{hasOptInDatauser}(?data, ?datauser) \leftarrow (2)$$

Based on (1) and (2), we have a feasible set of $\text{ABox}$ instances with $\text{data}$, $\text{purpose}$, and $\text{datauser}$ combinations of an attribute set that was permitted from the original $\text{dataowner}$ to allow a particular type of $\text{datauser}$ to ask for a $\text{data}$ set with a permissive $\text{purpose}$. When a server 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, $\text{action}$ and $\text{obligation}$, to complete the FIPs' privacy protection criteria. An ordinary data user is allowed to ask a query service with $\text{action} = \text{read}$ at the $\forall \text{P}$. The other actions, such as $\text{deletion}$ or $\text{modify}$, 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 data retention period or for a data owner updates his or her own profile data.
3.2 Data Request Services
A server declares its privacy policy in P3P before a data owner’s data is collected. Once a user accepts a server’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 $FS = input(?du, ?r, ?p)$, 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.

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

In this semantic privacy-preserving model, we propose the VP infrastructure to allow a server in each data source to offer its FPP at the VP to enforce FPPC. FPP in each data source is shown as $K = O + R$, where ontology $O = (T, A)$ and rule $R = (RS, F)$. At the VP, 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 VP 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 $ABox$ and to integrate the $RS$ with the updated data only once.

4. ONTOLOGY MAPPING AND MERGING
A merged ontology come from mapping and alignment that provides data integration services. In particular, data integration through ontologies, such as LAV is possible for multiple servers if a mapping language $ML$ provides a semantic mapping description between the $GS$ and the underlying $LS$ for each server [14]. In LAV, the relationships between the $GS$ and the $LS$ are established by making LAV assertions. Every assertions has the form $Q_{LS} \sqsubseteq Q_{GS}$, where $Q_{LS}$ provides the views of the conjunctive query (CQ) over the local schema $GS$ for each data source, and $Q_{GS}$ is a CQ over the global schema $GS$ at the VP.

A CQ for $Q_{LS}$ can be defined as a privacy-aware authorized view of each server so we do not disclose any non-shareable data to the VP whenever each server 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 query the relational database. This problem was previously investigated in [34].
A mapping can be shown as \((uid,e_1,e_2,n,\rho)\), where \(uid\) is a unique identity for the mapping, \(e_1\) and \(e_2\) are entity names, such as class or property, and in the vocabulary of \(O_1\), \(O_2\), \(n\) is a numeric confidence measure between 0 and 1, and \(\rho\) is a relation such as subsumption (\(\sqsubseteq\)), equivalence (\(\equiv\)), or disjointness (\(\perp\)) between \(e_1\) and \(e_2\) [23].

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 \(O\), to define the FIPs’ privacy protection criteria (see Figure 5) that 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 \(\mathcal{ML}\) provides semantic mappings for each entity \(e \in GS\) at the \(\forall P\) to the corresponding entities \(e_i \in LS_i\).

A perfect ontology alignment obtained via a mapping \((uid,e_1,e_2,n,\rho)\) and merging between \(T_i\) in \(O_i\) and \(T_j\) in \(O_j\) satisfies the following conditions:

- \(e_i \in T_i\) and \(e_j \in T_j\) entity names are either defined for describing the root class names which corresponding to the privacy protection concepts, such as purpose, action, datauser, data, and obligation or 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 1.

- \(\rho\) is either equivalence (\(\equiv\)) or subsumption (\(\sqsubseteq\)) between entity names of \(T_i\) and \(T_j\) schemas. In an equivalent (\(\equiv\)) case, we can find a pair of one-to-one corresponding entity names for \(e_i \in T_i\) and \(e_j \in T_j\) 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 \(e_i \in T_i\) and \(e_i \subseteq e_j \in T_j\), and vice versa.

### 4.1 Perfect Ontology Alignment

A mapping can be shown as \((uid,e_1,e_2,n,\rho)\), where \(uid\) is a unique identity for the mapping, \(e_1\) and \(e_2\) are entity names, such as class or property, and in the vocabulary of \(O_1\), \(O_2\), \(n\) is a numeric confidence measure between 0 and 1, and \(\rho\) is a relation such as subsumption (\(\sqsubseteq\)), equivalence (\(\equiv\)), or disjointness (\(\perp\)) between \(e_1\) and \(e_2\) [23].

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 \(O\), to define the FIPs’ privacy protection criteria (see Figure 5) that 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 \(\mathcal{ML}\) provides semantic mappings for each entity \(e \in GS\) at the \(\forall P\) to the corresponding entities \(e_i \in LS_i\).

A perfect ontology alignment obtained via a mapping \((uid,e_1,e_2,n,\rho)\) and merging between \(T_i\) in \(O_i\) and \(T_j\) in \(O_j\) satisfies the following conditions:

- \(e_i \in T_i\) and \(e_j \in T_j\) entity names are either defined for describing the root class names which corresponding to the privacy protection concepts, such as purpose, action, datauser, data, and obligation or 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 1.

- \(\rho\) is either equivalence (\(\equiv\)) or subsumption (\(\sqsubseteq\)) between entity names of \(T_i\) and \(T_j\) schemas. In an equivalent (\(\equiv\)) case, we can find a pair of one-to-one corresponding entity names for \(e_i \in T_i\) and \(e_j \in T_j\) 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 \(e_i \in T_i\) and \(e_i \subseteq e_j \in T_j\), 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 [20]. The computation complexity of answering SWRL-based queries might be undecidable regarding the verification of rights access permission unless these policies satisfy the DL – Safe conditions [29].

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 modelled 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 [31].

For each data request query service, a perfect mapping language \(\mathcal{ML}\) provide the semantically linking of an entity name \(c \in GS\) in the datalog rule at the \(\forall P\) to the entity name \(e_i \in LS_i\) in the datalog rule at serveri, where \(LS_i\) is the TBox of \(O_i\), and \(c\) is a class or a property name. If there does not exist an \(e_i \in TBox\), in a subtree of the \(LS_i\) on the same layer as \(c \in TBox\) in the global tree of \(GS\), then we can recursively find a superclass or superproperty of \(e_i\) with \(\sqsubseteq e_i\) as the corresponding entity name, with a confidence measure of \(n = 1\).

To successfully fulfill the semantically linking of any entity name \(c \in GS\) via \(\mathcal{ML}\), 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 \(LS_i\). But we still allow the designer to use different entity name string, \(e_i \in LS_i\), below the root layer of each local schema and to have an entirely different underlying subtree structure. We use Promptontology mapping algorithm first to synchronize the entity names between \(LS_i\) 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 \(\mathcal{ML}\) to map the name of a class entity \(c \in GS\) to one of the equivalent local ontology schema’s class entity name in a deeper subtree, say \(c_j \in LS_j\), i.e., \(c \equiv c_j\) in the datalog rule’s conditions of each data request service. When the class semantics for \(c\) is \(c \sqsubseteq c_i\) in the \(LS_i\), i.e., we do not have a corresponding class \(c_i' \in LS_i\) on the same lower layer of a schema tree as \(c \in GS\). 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. 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 \(p \in GS\) is mapped to another equivalent property \(p_j \in LS_j\) for the associated datalog rule’s body conditions. Then property \(p \equiv p_j\) might be on a lower layer in the schema tree when compared with property \(p_i \in LS_i\). We still regard property \(p_i\) as feasible for its enforcement of the datalog rule on data sharing and protection. Finally, if we consider mappings for binding property and class from the aligning ontology schema \(GS\) to \(LS_i\) and \(LS_j\) to the
respectively.

Property $p \in GS$ with its domain class $dc$ and range class $rc$ that are mapped to property $p_i \in LS_i$ with its domain class $dc_i$ and its range class $rc_i$. For each data request service using a perfect mapping language $ML$, when $p \subseteq p_i$, we use property $p_i$. Otherwise, when $p \subseteq p_i$, we use property $p$ for the datalog rule $r_i$. When $dc \subseteq dc_i$ and $rc \subseteq rc_i$, we use class $dc_i$ and $rc_i$. Otherwise, when $dc \subseteq dc_i$ and $rc \subseteq rc_i$, we use class $dc$ and $dc_i$ for the datalog rule $r_i$.

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 property and 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 $VP$.

Example 1. In Figure 6, after we map and align two local partial ontology schemas, $LS_1$ and $LS_2$, into a merged partial ontology global schema $GS$, we receive a data request service with class $P_{212}$. In the purpose class $P$, $P_{111} \leadsto P_{211}$, but $P_{212} \in GS$ does not have a corresponding subclass in $LS_1$, since $P_{212} \subseteq P_{211}$. When a data request service asks for class $P_{212} \in GS$, mapping language $ML$ will map $P_{212}$ to $P_{111}$ for the datalog rule $r_i$ to query the $LS_1$.

5. PERFECT RULE INTEGRATION

In $FPPC$, we define an integrated rule set $\bigodot R_i = (\bigodot R_{S_i}, \bigodot F_i)$ to enforce data query and protection services in $\bigodot O_i$. In fact, an integrated rule set $\bigodot R_{S_i}$ is a part of $FPPC$ that was created by collecting the datalog rules, e.g., SQWRL queries, in the formal policies $FP_i$, from local servers. A datalog rule $r_i$ in the $R_i$ of $FP_i$ is shown as $^2$:

$$H \leftarrow B_1 \land B_2 \land \cdots \land B_n,$$

where $H$, the query results (or views) are expressed as SQWRL built-ins, such as $\text{sqwr1:select}$ and the rule antecedent $B_i$ 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.

A perfect rule integration is defined for the integration of any datalog rules as $\exists x_i \in R_{S_i}$ in $FP_i$, for the purpose of data sharing and protecting without causing conflicts with $\exists x'_i \in \bigodot R_i$, $\lambda_i \in \bigodot O_i$, i.e., conditions do not exist for $\exists x_i \models \lambda_i \Rightarrow \exists x_i' \neq \lambda_i$, or $\exists x_i \neq \lambda_i \Rightarrow \exists x_i' = \lambda_i$. Then, $\exists x_i \in \bigodot R_i$ in the $VP$ can be activated and mapped by the perfect mapping language $ML$ into $\exists x_i$ to enable a global data query and protection service of multiple servers.

$^2$This datalog rule is related to a $CQ$ of the form: $v_i \leftarrow \text{conj}(\{ f_i \})$.

Example 2. A rule $r'_i$, is one of the rules within the integrated rule set at the $VP$. It asks for a data set $?d$ with related obligations $?o$ under the feasible parameter input set $FS = (M_1,TMarketing6,Read2)$, where data user $M_1$ is a marketing staff with a purpose of achieving telephoning $TMarketing$. A rule $r'_j$ is mapped to a rule $r_i$ and a rule $r_j$ using the rule mapping processes when we have done an upward perfect ontology mapping, alignment, merging and a perfect rule integration. A downward perfect mapping language $ML$ operation maps the $r'_i$’s predicates, such as class, property to the corresponding predicates in a rule $r_j$ and a rule $r_j$ with $\text{Datauser}(M_1) \sqsubseteq \text{Datauser}(M_1)$, $\text{TMarketing}(TMarketing6) \sqsubseteq \text{Purpose}(TMarketing6)$. Therefore, real data query and protection services requested by a rule $r'_j$ are performed by a rule $r_i$ and a rule $r_j$.

A rule $r_j$ queries at the $O_j$:

$$\text{Datauser}(M_1) \land \text{TMarketing}(TMarketing6) \land \text{DatauserHasPurpose}(M_1,TMarketing6) \land \text{DatauserHasAction}(M_1,Read2) \land \text{hasOptInPurpose}(?d,TMarketing6) \land \text{hasOptInDataUser}(?d,M_1) \land \text{purposeHasObligation}(TMarketing6,?o) \rightarrow \text{sqwr1:selectDistinct}(?d,M_1,TMarketing6,Read2,?o)$$

A rule $r_j$ queries at the $O_j$:

$$\text{DatauserHasPurpose}(M_1,TMarketing6) \land \text{DatauserHasAction}(M_1,Read2) \land \text{hasOptInPurpose}(?d,TMarketing6) \land \text{hasOptInDataUser}(?d,M_1) \land \text{purposeHasObligation}(TMarketing6,?o) \rightarrow \text{sqwr1:selectDistinct}(?d,M_1,TMarketing6,Read2,?o)$$

Example 3. Under the data protection law, two hospitals, $A$ and $B$, have allowed to share their patients’ Electronic Health Records (EHRs) after patients give their consents for the medication purpose. A patient was hospitalized in the hospital $A$ for a surgery. After that, this patient went to the hospital $B$ for an outpatient medication. A physician in the hospital $B$ was authorized to query this patient’s shareable EHR at the $VP$ collected from hospital $A$ and hospital $B$'s RDB data sources. The vocabularies of partial ontology schemas for hospital $A$’s local schema $LS_A$, hospital $B$’s local schema $LS_B$, and the global schema $GS$ at the $VP$ are shown as Figure 7.

Hospital $A$ has the following terms as its ontology’s local schema $LS_A$ vocabularies:

Class: $\text{Clinic}$ and $\text{HealthData}$ with subclass $\text{SurgeryData}$ and $\text{HospitalizationData}$.

Property: create with domain class as $\text{Clinic}$ and range

[9]
Views use at the VP created from the hospital B local schema’s vocabularies are:

def(V6Person) = Patient

def(V7HealthCenter) = Hospital

def(V8PatientData) = HealthRecord

def(V9OutPatientData) = HealthRecord ∧ ∀ hasMedType.Patient Data

def(V10beMedicated) = beCured

def(V11owns) = hasHealthRecord

A physician queries a patient’s surgery record at the VP by using a merged global ontology schema based on LAV query rewriting instead of directly requesting each hospital. An original datalog-based SQWRL rule for a query q at the VP is shown as:

`select(?x, ?r) ← sqwrl: select(?x, ?r)`

Query rewriting of the q in terms of two CQs, e.g., q\textsubscript{A} and q\textsubscript{B}, uses views defined at the VP:

\[
V6\text{Person} ∧ V10beMedicated ∧ V11owns ∧ V9OutPatientData ∧ V5create ∧ V1owns → sqwrl: select(?x, ?r) ← (q\textsubscript{A})
\]

Use LAV approach to define each class and property in these two hospital local schemas as views in terms of the global schema’s vocabularies shown as follows:

T \sqsubseteq ∀ create.Clinic

T \sqsubseteq ∀ create˜.HealthData

Hospital B has the following terms as its ontology’s local schema LS\textsubscript{B} vocabularies:

Class: Person, HealthCenter, and PatientData with sub-Class OutPatientData

Property: own, beMedicared with their respective domain and range class are shown as follows:

T \sqsubseteq ∀ own.Person, T \sqsubseteq ∀ Own˜.PatientData.

T \sqsubseteq ∀ beMedicared.Person,

T \sqsubseteq ∀ beMedicared˜.HealthCenter.

The VP offers the following vocabularies:

Class: Patient, Hospital, Surgery, and HealthRecord

Property: beCured, hasHealthRecord, generate with their respective domain and range class are shown as follows:

T \sqsubseteq ∀ beCured.Patient, T \sqsubseteq ∀ beCured˜.Hospital

T \sqsubseteq ∀ hasHealthRecord.Patient

T \sqsubseteq ∀ hasHealthRecord˜.HealthRecord

T \sqsubseteq ∀ generate.Hospital

T \sqsubseteq ∀ generate˜.HealthRecord

Use LAV approach to define each class and property in these two hospital local schemas as views in terms of the global schema’s vocabularies shown as follows:

Views use at the VP created from the hospital A local schema’s vocabularies are:

\[
\text{def(V1Clinic)} = \text{Hospital}
\]

\[
\text{def(V2HealthData)} = \text{HealthRecord}
\]

\[
\text{def(V3SurgeryData)}
\]

= HealthRecord ∧ ∀ hasMedType.Surgery

A physician queries a patient’s surgery record at the VP by using a merged global ontology schema based on LAV query rewriting instead of directly requesting each hospital. An original datalog-based SQWRL rule for a query q at the VP is shown as:

`select(?x, ?r) ← sqwrl: select(?x, ?r)`

Query rewriting of the q in terms of two CQs, e.g., q\textsubscript{A} and q\textsubscript{B}, uses views defined at the VP:

\[
V6\text{Person} ∧ V10beMedicated ∧ V11owns ∧ V9OutPatientData ∧ V5create ∧ V1owns → sqwrl: select(?x, ?r) ← (q\textsubscript{B})
\]
Above $q_a$ query is corresponding to a query as:

$$B : \text{Person}(p) \land B : \text{beMedicated}(p, c) \land B : \text{own}(p, d) \land A : \text{SurgeryData}(sd) \land A : \text{create}(h, ?hd)$$

$$\rightarrow \text{sqrl: select}(?, sd)$$

6. **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 $GPS = (O_i, \bigcirc R_i)$ at the $VP$. If $q(x)$ is a $\exists Q$ over $O_i$ at the $VP$ and $q_a(x)$ is a $\exists Q$ over $O_i$, using LAV source descriptions from a data server, then $\forall x. q(x) \rightarrow \exists Q_i(x)$.

In [15], authors showed that when a query has a finite number of *maximally contained conjunctive rewritings*, then 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 $O + R$, 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 $O + R$. A perfect ontology merging and a rule integration ensure the soundness and completeness of data sharing and integration in the semantic privacy-preserving model. This will be briefly shown as follow:

### 6.1 [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) $r_i \in \bigcirc R_i$ at the $VP$ instead of using a direct query service as rules (queries) $r_i \in R_i$ in each server, $\forall i$.

**Theorem 1.** [Soundness] After a perfect ontology alignment and rule integration with $\text{FPPC}$, $\exists GPS = (\bigcirc O_i, \bigcirc R_i)$ at the VP, Under a particular feasible parameter input set $FS_i$, if $\lambda_j \in O_i$ is protected by a $\text{FPPC}$ at each server, $\forall i$, i.e., $\forall i, r_i \in R_i \Rightarrow \lambda_j$, then $r_i' \in \bigcirc R_i \neq \lambda_j$ for the same $FS_i$, where $\lambda_j$ is a protective data set in $O_i$.

**Proof.** (Sketch) If $q(x)$ is a query over $\bigcirc O_i$ at the $VP$ and $q_a(x)$ is a query over $O_i$ in a server, then we need to prove the statement $\forall x. q(x) \rightarrow \exists Q_i(x)$. This statement is equivalent to the original argument: If $r_i \in R_i \neq \lambda_j$, then $r_i' \in \bigcirc R_i \neq \lambda_j$. The $\exists Q$ $q(x)$ is a query containment of datalog rule $r_i'$, and the $\exists Q$ $q_a(x)$ is a query containment of datalog rule $r_i \in R_i$. The statement $\forall x. q(x) \rightarrow \exists Q_i(x)$ is true because the local as view (LAV) schema mapping only allow the protected concept $\lambda_j$ in each server 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: $\exists r_i \neq \lambda_j \Rightarrow \exists r_i' \neq \lambda_j$.

### 6.2 [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) $r_i' \in \bigcirc R_i$ at the $VP$ instead of using a direct query service as rules (queries) $r_i \in R_i$ in each server, $\forall i$.

Figure 7: A partial ontology for Electronic Health Record (EHR) sharing and privacy protection.
The role-based access control (RBAC) model is used to enforce the access control policies with a static role assignment for a stand-alone system. It is therefore not useful for solving the privacy protection problem. In fact, the RBAC model did not consider the prime elements of the FPs, so it is not intended for a privacy protection problem. In [32], the \( \text{UCON}_{a} \) might be useful for the privacy protection problem, but it did not explicitly allow the data sharing and protection in multiple sites.

The EFAF access control model is an extension of the FAF that provided the solution for privacy protection [22] [24]. This is the closest method to our solution, but its privacy protection control is more on the logic program and less on the ontology schema for the structure data modeling. This also prevents the data sharing and protection in multiple sites. The other similar models for enforcing the enterprise privacy protection go to the following EPAL [25] [35]. OASIS XACML is a policy language for privacy and digital rights protection. But it is an XML-based policy language so the policies based on XACML possibly might have ambiguous semantics that prevent using a flexible policy combination in multiple servers [1].

8. CONCLUSION AND FURTHER STUDY

We propose a semantic privacy protection model which encompasses and extends the existing works on data sharing and integration. We intend to solve the privacy protection problem to provide data sharing and integration in multiple servers by using one of ontology and rule language combinations, e.g. SWRL. Another OWL2 combination will be considered for the future [17]. This can be extended to a modular reuse of ontologies for data sharing and protection in the cross-domain cloud computing environment [16].

The perfect ontology alignment through ontology mapping and merging creates a global ontology schema at the V\( \mathcal{P} \) by integrating multiple local ontology schemas from different data sources. 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 V\( \mathcal{P} \). In fact, a datalog rule is considered as a conjunctive query, which provides data query and protection services in each server.

However this perfect ontology alignment is impossible without the requirements of using same ontology schema for the root layers for multiple servers with the LAV schema mapping. We face another policy hidden conflict challenge if background default policy assumptions are vary between different servers. All of these need further study.

Finally semantics-enabled policies are combined together at the V\( \mathcal{P} \), so we simplify the data sharing and protection services. But the soundness and completeness criteria are still preserved for data sharing and integration purposes. This supports the trustworthiness of a policy combination for multiple servers.

Acknowledgements

This research was partially supported by the NSC Taiwan under Grant No. NSC 99-2221-E-004-010.
9. REFERENCES


