

# An Ontology Regulating Privacy Oriented Access Controls

Maherzia Belaazi<sup>(✉)</sup>, Hanen Boussi Rahmouni, and Adel Bouhoula

Digital Security: Research Unit, Higher School of Communication of Tunis,  
University of Carthage, Tunis, Tunisia  
{maherzia.belaazi, hanen.boussi, adel.bouhoula}@supcom.tn

**Abstract.** Access Control is one of the essential and traditional security weapons of data protection. In open and complex environments such as the Internet or cloud computing, the decision to grant access to a resource must ensure a secure management with a specific attention to privacy and data protection regulations. In recent years, many access control models and languages were proposed. Despite increasing legislative pressure, few of these propositions take care of privacy requirements in their specifications. In this paper we propose to enforce privacy compliance in access control policies. Based on a semantic modeling approach, specifically formal ontology, we will try to incorporate data protection legislation requirements in policies specification and implementation. This aims to abstract the complexity of legal requirements expression and to facilitate their automation and enforcement at execution level. Indeed, at run time, the interoperability of diverse information and the reference to the text law are addressed in a novel manner.

**Keywords:** Security · Access control policies · Privacy policies · Legislation · Semantic web · Ontology

## 1 Introduction

Access control is defined as one of the most popular security tools ensuring that every access to a system and its resources is controlled and only those access that are authorized can take place. It is now evolving with the complex environments that it supports. In recent years for different aims, many and different access control models and languages have been proposed. In open and complex environments such as the Internet or cloud computing, the decision to grant access to a resource must ensure a secure management with a specific attention to privacy and secondary usage regulations [1]. Besides, in order to ensure an efficient data access decision, the issue of interoperability between access policies related to different usage scenarios must be addressed. For example, in the context of cloud computing, we must ensure that the entity *requestor* and the entity *provider* of an access control policy context share the same meaning or are equivalent [2]. In the scope of the previously mentioned requirement, we propose in this paper,

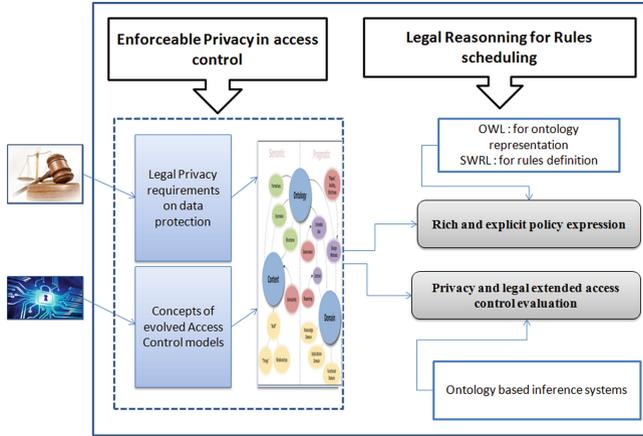
to exploit technologies such as OWL (the ontology web semantic language), SWRL (Semantic Web Rule language) along with other languages forming the semantic web stack. The driver for our choice is the demonstrated powerful expressiveness capabilities in existing privacy protection related works [3, 4].

In philosophy, the term ontology means a systematic account of existence [5]. The term has been widely adapted for formal use in the Artificial Intelligence domain and other computer science domains where knowledge representation is required. According to these domains, an entity can “exist” only if it can be represented. A formal ontology is a body of formally represented knowledge that is based on an explicit formal specification or conceptualization of concepts and the relationships that could exist between them [6]. Since open environments are creating a growing demand for sharing data as well as its semantics, ontologies are becoming increasingly essential for most computer applications. By sharing ontologies and knowledge bases, distributed environments can easily communicate to exchange data and thus make their transactions interoperate independently of their enabling technologies [7].

In order to ensure secure access to its resources, an organization defines a set of policies. A policy is a statement that defines and controls the behavior of a system. More specifically, an access control policy defines a set of conditions, which are evaluated to determine whether a set of actions may be performed on a resource. For example, an access control policy could state that only doctors (i.e. the condition) have the right to modify (i.e. the action) the patient’s medical records (i.e. the resource). An access control policy acts as both, a declarative behaviour system (express in explicit and non ambiguous manner the requirements of control) and, a decision-support system (base of access control inference system). Semantic Web languages allow access policies to be described over heterogeneous data domain and promote common understanding among participants who might not use the same information model [2].

In our research, we focus on the requirements for sensitive data protection driven by legislation. We look on how to incorporate these requirements in an access control model for an open and dynamic environment such as the cloud. Our research is aiming to help enhancing privacy compliance when sharing private data across diverse states or countries where legislation could be different. Our approach (Fig. 1) is based first on bridging the gap between high level legislation on data protection and operational level controls by means of semantic modeling of important concepts. Second, this approach is also based on the use of inference systems on top of the modeled concepts in order to provide decision support on how to handle personal data.

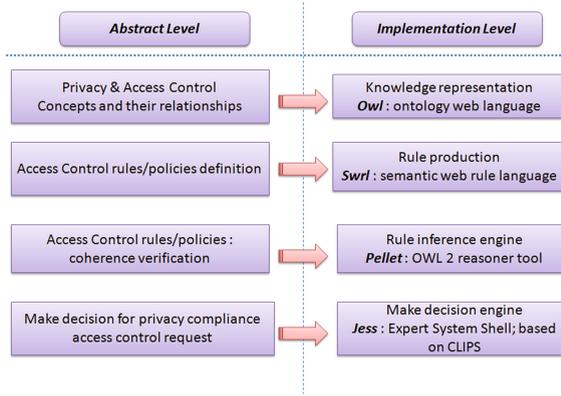
In literature, and in order to deal with privacy compliance, traditional access controls (like role based access control or attribute based access control) [8, 9] have been extended by introducing the purpose concept. We find also in literature, different proposed ontologies [3, 4] (and associated semantic rule execution engines) that represent access control concepts in addition to other legal concepts (like the consent). In our perspective, these propositions lack the explicit expression of the law reference (the used text law and the authority location



**Fig. 1.** Toward a secure and private access control requirements formalization

proposing this law). Certainly, using ontology helps to ensure interoperability between access control actors (access requestor, access provider and resource’s owner) by providing the same shared conceptual semantic model. However, we need also to provide mechanisms for a privacy compliance access decision at runtime: which prior rule to apply? which law to apply if we deal with multi-authority (access requestor, access provider and resource’s owner: when they belong to different jurisdiction areas)? So in our contribution, we will present and exploit knowledge about the regulation reference to conclude about which law to apply. Besides, we will formulate privacy requirements (purpose, consent, data privacy status, the authority obligation) (see Sect. 3.2) and use them as attributes to calculate the *legal strength* of an access rule in order to guide a *partial* scheduling of the access control rules (see Sect. 3.3). We believe that such legal reasoning will help to ensure a privacy compliant access decision in a distributed multi-authority environment. In order to achieve our goal, we proceed in parallel axes. The first axe is the abstraction level. In this level we formally define privacy and access control concepts and their relationships. We also formally define how rules and policies should be expressed in generic and fine-grained manner.

The remainder of this paper is organized as follows. Section 2 settles the methodology we follow in order to build and validate our ontology. In Sects. 3 and 4, we first define different concepts describing access controls and data protection requirements which we merge in a single extensible ontology. Secondly, we present formally how to express and evaluate an access control policy incorporating legal requirements extracted from data protection legislation. The formal presentation will be the base of rules and inference system definition on top of the proposed ontology. In Sect. 5, we experiment our model in a case study by instantiating the generic ontology in a specific domain (medical domain). Finally, we conclude by future work and perspectives.



**Fig. 2.** Privacy requirement incorporation in access control: mapping between abstract and implementation levels

## 2 The Privacy Ontology Construction and Validation

Based on some literature lectures about ontology creation and maintenance [10–13], we define a methodology to follow in order to establish our ontology. The proposed methodology is divided in three principal phases: (Subsect. 2.1) ontology’s purpose specification (Subsect. 2.2) ontology’s construction (Subsect. 2.3) ontology’s validation.

### 2.1 Ontology’s Purpose Specification

Before starting the ontology’s construction, we must explicitly answer the following questions [12] as follows.

- *What is the domain that the ontology will cover?* In this paper, we try to define a conceptual model of an access control scenarios. These scenarios invoke the policies definition, the request expression and the access decision assignment. The decision related to a request must be secure and privacy compliant too. In this spectrum, our ontology must cover, in a first stage, privacy requirements extracted from legislation and also from access control management frameworks applied to generic domains. In a second stage, some application domains could be introduced to experiment previous generic domains. We can experiment, as an example, the medical domain.
- *For what purpose we use the ontology?* Our ontology concepts will help us to establish closed real world reasoning. It will enable us (using clear rules and inference systems) to perform secure access control decisions while achieving: (i) complying with privacy requirements (ii) resolving the semantic interoperability issue on different concepts used by different actors across heterogeneous domains.

- *What type of questions should the ontology provide answers to?* This ontology could help the user to check an access request defined by a set of parameters (requester attributes, context specification, target attributes) allowed or prohibited? Is this privacy compliant? Is it a legal access request? In the case of positive permission for a resource access, does it require some obligations to be filled? In case of denial of the access request, because of privacy non compliance, what are the requirements to be reported to the access requestor?
- *Who will use and maintain the ontology?* The access control administrator should maintain this ontology. They need the help of law experts to understand emerging legislation requirements.

## 2.2 Ontology Construction

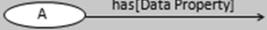
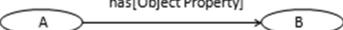
In order to construct our ontology [10–12], we propose to define in the following order. Firstly, we define a list of concepts (the Classes). Secondly, we specify the list of concept’s properties. Thirdly, we define the possible hierarchical relations between defined concepts, as far as the list of conceptual relations between them. Finally, we instantiate the general concepts by defining the individuals.

## 2.3 Ontology Validation

In order to validate our ontology [13], we propose to proceed in this order. Firstly, we ensure that the structure and the conceptualization is valid. This encloses the fact to check that no redundancy is available and the logic coherence is respected. Secondly, we do the functional validation. This includes the validation of the ontology’s purpose definition conformity. This will be achieved using specific questions (ontology interrogation using queries) and the evaluation of the ontology based system’s answers.

Our approach uses a free, open source ontology editor framework named the Protg (version 4.3) [14]. We use the OWL-DL Ontology Language: (for knowledge representation) [15,16] and SWRL (Web rule language to express queries on top of the ontology) [17]. The OWL model we have built is used to represent privacy requirements extracted from the most internationally recognized privacy legislation in the context of secure and privacy oriented access control evaluation and decision making. OWL allowed us to model the conceptual domain of “access control policies” and “data processing” obligations for the usage of private personal information as a hierarchy of classes, subclasses and a hierarchy of data and object properties to represent the relationships between them. Classes and objects properties could easily be used to express privacy requirements. Additional class expressions including restrictions, Boolean expressions, enumerated classes and value partitions were also useful. SWRL allowed us to express privacy policies in order to ensure a privacy preserving access control decisions. In order to get clearer vision we will provide ontology graph (nodes and edges) following the OWL paradigm (Fig. 3).

In the next sections, we choose to work in stages. In a first stage, we build an ontology that models access control policies. In a second stage, we establish

GraphSymbol	Meaning
	Presents a Class ( a real World Concept).
	Presents an Individual (an instantiation of a class).
	Indicates Class A has Data Property (example hasAge property equal to 15).
	Indicates Class A has Object Property with range Class B.
	Indicates Class A is a subclass of Class B.

**Fig. 3.** Ontology graph symbols

an ontology that models Personal Data Privacy requirements for a legislation compliant access control. “Privacy Obligations” have been extracted from international privacy laws [18, 19]. We have done a comparative study (some privacy’s subject based thesis [20, 21]) that concluded common mandatory legal concepts to express and check while dealing with privacy preservation. Finally, we will merge the two previous ontologies in order to incorporate privacy obligations while expressing an access control policy or while evaluating an access control request.

### 3 An Ontology-Based Description of Privacy Obligations in Access Control Policies

In this section, and following the previously defined methodology, we define an ontology that represents incorporating privacy obligations in access control policies. We start by defining an ontology for access control then we extend this ontology by privacy requirements. Indeed, we provide some formal definitions that will be the base for next stage which looks at access control reasoning operations.

#### 3.1 Formalization of the Access Control Model

Security policies constitute the core of systems protection. They are made of a set of instructions specifying allowed and prohibited actions. Access control policies are examples of security policies defining which subject (requestor of a resource) could or not invoke an action (operation) on an object (resource). Access control is now evolving with complex environments that it supports. In recent years and for different aims, many access control models and languages have been proposed.

In order to build a knowledge base about access control policy specification and requirements, we study and analyze a set of novel and emerging models in the access control field to address the new needs of today's systems [8, 9, 22–24]. We studied also the OASIS standard access control language: the XACML [25]. In fact, XACML is a widely adopted policy language standard that has proven efficiency in the enforcement of policies at operational level [26]. Based on the previous studies and analysis we define our list of concepts and their relations. (Fig. 4)

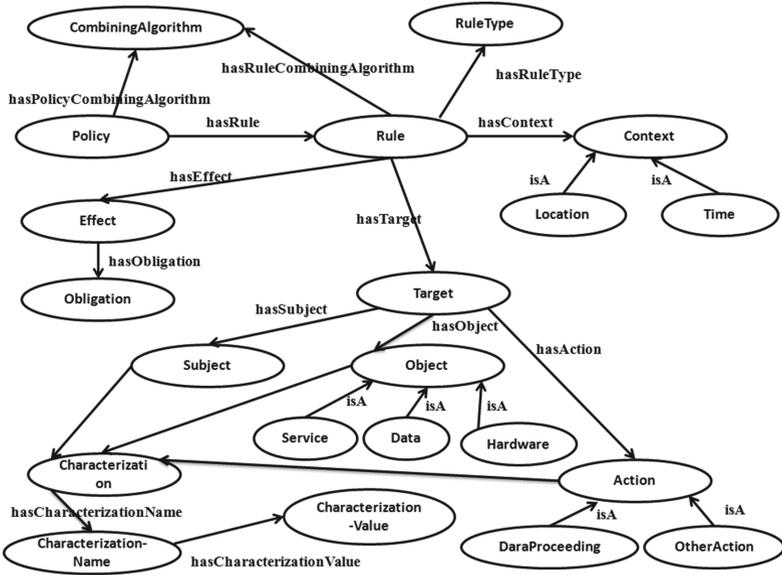


Fig. 4. Privacy oriented access control ontology: part 1

We define as relevant list of access control requirements related concepts the list below:

- **Policy:** Defines a set of rules expressing how an organization will manage, protect their resources.
- **CombiningAlgorithm:** Defines the way of merging the different rules of one policy. It defines also how to merge a set of policies and which rule to use.
- **Rule:** Defines the characteristic (including the identity and/or role) of a subject who is authorized to perform an action on a resource and under which conditions, the requested authorization is given or prohibited.
- **RuleType:** Defines the type of a rule; it could be a user preference or organizational policy.
- **Effect:** Expresses the response to an access request: deny or allow or undetermined

- **Obligation:** Expresses the actions that a subject must perform following a positive access response. (example: duration of information retention) Or Express the missing requirements that a subject must perform following a negative access response. (Example: consent of data owner).
- **Target:** Defines the set of associated subject object and action. Optionally we can invoke a receiver if the action invoked needs them. (E.g. a doctor can share patients data with an-other doctor in order to get a second opinion, here the information about the other doctor could be relevant for access control decision). Each Target element is described by its **characterization** which is a couple of  $\langle characterization_{name}, characterization_{value} \rangle$ . For instance, a Subject could have age as  $characterization_{name}$ .
- **Action:** Defines an operation (example: read, modify, collect, store, share, forward ....) required on some resources.
- **Object:** Defines a resource. It could be a service (for example the wifi access), a data (for example medical tests result) or hardware (for example a printer).
- **Subject:** Defines who requested an action on an object. A subject is identified by a set of his characteristics.
- **Context:** Defines some constraints on location and time.
- **Location:** Defines the location associated to an object.
- **Time:** Defines the time of the request (start time and end time implicitly the duration of the request)
- **Service:** For example: the use of a wifi connection.
- **Data:** Defines a kind of Resource, more precisely, the hosted information on the sys-tem/organisation
- **Hardware:** Defines a kind of Resource like the printer.
- **DataProcessing:** Defines a kind of action on data
- **OtherAction:** Defines any other kind of action on any other kind of resource different from data. (For example, configure the printer).

The main relations between previous detailed concepts are listed below. A relation is defined in the same manner as functions ( $f : Domain \rightarrow Range$ ). A semantic relation is defined between a “domain of concepts” and “range of other concepts”:

- **hasRule:** Our domain is Policy and our range is Rule. So in simple words, a Policy is com-posed by a set of Rule.
- **hasPolicyCombinigAlgorithm:** Our domain is Policy and our range is CombiningAlgorithm. This property describes how to manage a set of policies (the order of execution? the combining manner?)
- **hasRuleCombinigAlgorithm:** Our domain is Rule and our range is CombiningAlgorithm. This property describes how a policy manages a set of rules. In which way it chooses a rule to apply if many rules respond to a request?
- **hasType:** Our domain is Rule and our range is Type. Each rule must be classified to a specific type. It could be a rule imposed by the organisation, or a rule imposed by the user. For example, in cloud computing the provider of the service has its secure policies and the customer could dictate some of its preferences.

- **hasTarget:** Our domain is Rule and our range is Target. Each defined rule has a Target.
- **hasContext:** Our domain is Rule and our range is Context. Each defined rule may have contextual constraint.
- **hasEffect:** Our domain is Rule and our range is Effect. Each defined rule has an effect.
- **hasObligation:** Our domain is Effect and our range is Obligation. Some effects are associated to some obligations that must be filled. (In case of “allow” effect, an obligation could be the fact to respect a duration of the retention.)
- **hasAction:** Our domain is Target and our range is Action. A target is defined by an Action.
- **hasObject:** Our domain is Target and our range is Object. A target is defined by an Object.
- **hasSubject:** Our domain is Target and our range is Subject. A target is defined by a subject.

### 3.2 Formalization of Privacy Requirements

Privacy is the right of individuals to decide for themselves when, where what how and who can extent, disclose or use their personal information. Privacy principles (Fig. 5) have been developed thanks to many legislation. Internationally, the OECD (Organization of Economic Cooperation and Development) [18] provides the most commonly used privacy framework, they are reflected in existing and emerging privacy and data protection laws, and serve as the basis for the creation of leading practice privacy programs and additional principles. The XACML<sup>1</sup> OASIS Standard describes a profile for expressing privacy policies. This profile extracted Privacy requirements from the OECD guideline. Based on the OECD principles, the Privacy XACML profile and some other international legislation [19], we list the essential privacy obligations to respect in our ontology:

- **Collection Limitation Principle:** The collection of personal data requires the knowledge or the consent of data subject (the data owner). Such personal data should be obtained by lawful means.
- **Purpose Specification Principle:** The purposes for which personal data are collected should be explicit.
- **Use Limitation Principle:** Personal data should not be disclosed, made available or otherwise used for purposes other than those specified in accordance with previous paragraph except with the consent of the data subject; or by the authority of law.

According to the previously listed and explained privacy principles, we define below a list of our ontology concepts and their relationships. (Fig. 6) Starting by the list of concepts:

<sup>1</sup> XACML privacy profile is a new profile proposed by the last XACML version 3.0 (at the time of writing this paper).

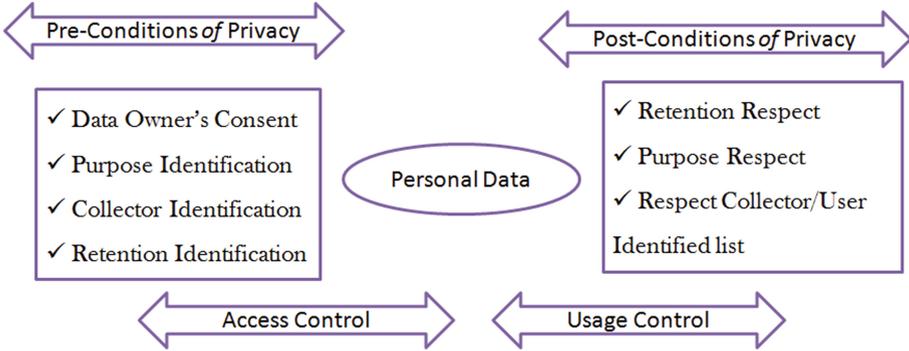


Fig. 5. Essential privacy conditions in access and usage control

- **data:** Defines a category of object that could be question of access request.
- **dataOwner:** Defines the owner of the data.
- **dataController:** Defines the controller of the data who is legally responsible to enforce the privacy. For example, in case of cloud computing the cloud provider who hosts the data.
- **dataRequestor:** Defines the requestor of the data. It is equivalent to the subject concept in the previous defined ontology.
- **dataCategory:** Defines the category of the data. We classify data in two main categories: non personal data and personal data.
- **NonPersonalData:** the non personal data category defines data that cannot be used to identify a person. Example, a person’s hobbies.
- **PersonalData:** Personal data (as defined in EU directive or PII<sup>2</sup> in USA laws) is any information that can be used on its own or with other information to identify a person. For example, a card identification number (card-Id).
- **NonSensitivedata:** For example, age name ...
- **SensitiveData:** For example, religion medical information ...
- **Legislation:** Defines the legislation restricted to the location of the resource.
- **Purpose:** The reason for which something is done or created or for which something exists.
- **ProcessingType:** Equivalent to action type; data processing type
- **Collect:** It is a data processing type.
- **Share:** It is a data processing type.
- **Disclose:** It is a data processing type.
- **Modify:** It is a data processing type.
- **legalRequirement:** It is an optional requirement defined by “data” location and used for “data processing”
- **Consent:** It is a category of legal requirement.
- **Anonymity:** It is a category of legal requirement.
- **Notification:** It is a category of legal requirement.

<sup>2</sup> Personally identifiable information.

- **Legislation:** the associated legislation of one legal requirement.
- **legalStrength:** Defines the power of a law. In fact, all legal texts do not have the same value. For example, federal and state laws are not equivalents.
- **Reference:** the reference of a law. (For example, international convention)
- **textLaw:** Defines explicitly the referenced legal text law. (For example: constitution, laws, decrees, orders, proceedings ...)

The previous list of concepts has different relations (Fig. 6). These relations are used in the access control case expression. For example, the **data** concept has centric relations:

- **hasDataController:** Defines the controller of the data.
- **hasDataOwner:** Defines the owner of the data.
- **hasDataRequestor:** Defines the requestor of the data.
- **hasDataCategory:** Defines the category of the data.
- **hasProcessingType:** Defines the processing type on a data.

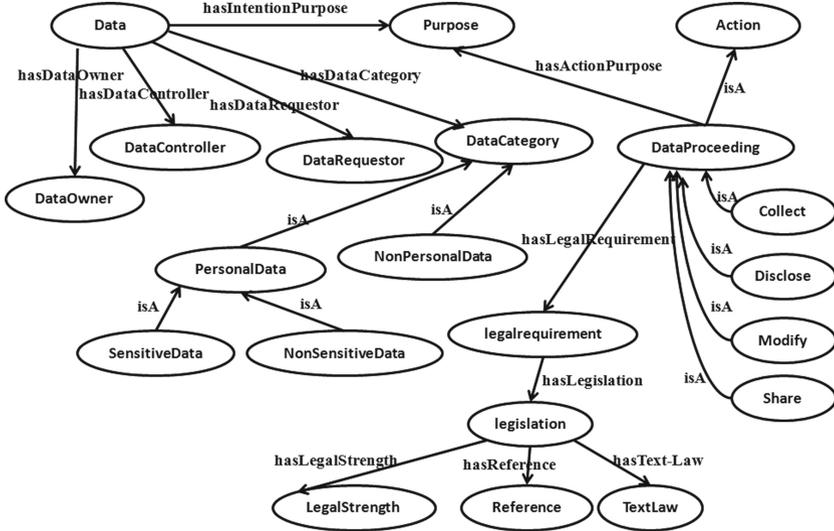


Fig. 6. Private access control ontology: part 2

### 3.3 Towards a Privacy Aware Decision Making Engine

In next paragraphs, we give some formal definitions of private secure rules. These definitions will be used in next stage for the reasoning engine construction. (Make decision engine explained in Fig. 2).

**Formal definition (1):** Express a security policy based on a set of access control policies

A security policy is a set of access control policies ( $P_i$ ) and the manner to merge them ( $CA$ : combining algorithm). The utility of a  $CA$  is to manage some particular cases. The first case, if no rule is applicable, which decision to make? For example, if “deny-overrides” combining algorithm is set, the decision will be “no”. The second case, if more than one rule is applicable, which rule to choose? For example, if “first-applicable” combining algorithm is set, the first rule is to be evaluated.

$$SP = (\{P_i, i \in IN\} \times CA)$$

$CA \in \{\text{deny-overrides, permit-overrides, first-applicable, only-one-applicable}\}$

- *Deny-overrides*: returns deny if any evaluation returns deny. Otherwise, permit is returned.
- *Permit-overrides*: If any rule evaluates to permit, then the result is permit. If any rule returns deny and all other rules evaluate to not applicable the result is deny. If all rules evaluate to not applicable, the result re-mains as not applicable.
- *First-applicable*: Returns the first result of a rule evaluation that is different from not applicable. If all rules return not applicable, this response is returned.
- *Only-one-applicable*: If only one policy is applicable its result is returned. If more than one applies the result is indeterminate. If no rule is applicable, the result is not applicable.

In this paper, we propose a general specification for one policy a set of rules evaluated according to a combining algorithm  $CA$  (the same list of  $CA$  defined previously):

$$\begin{aligned}
 P &= (\{Rule_j, j \in IN\} \times CA) \\
 Rule_j &= Req_j \longrightarrow Resp_j \times [oblig_j]^*; j \in IN \\
 Req &= (s_j \times a_j \times o_j \times [c_j]^*); j \in IN \\
 Resp &\in \{yes, no, undetermined\}
 \end{aligned}$$

In our scope, a rule ( $Rule_j$ ) is composed by two parts. The first part is the request ( $Req_j$ ) (next definition explains the access request). The second part is the response ( $Resp_j$ ). A response values could be “yes”, “no” or “undetermined”. In our definition, the response could be optionally associated to a set of obligations ( $Oblig_j$ ).

In case of “yes” response, the obligation, could be for example, “the duration of retention”.

In case of “no” response, the obligation, could be for example, “the consent of the data owner”.

**Formal definition (2):** *Express an access request*

A request  $Req_i$  is defined by several fields:  $s$  is the “subject”,  $o$  is the “object”,  $a$  is the “action” and  $c$  is the “context”. “ $s_j$ ” field describes the subject who’s the requestor of access. The subject is defined by a list of attributes and their values.

For example,

$doctor = \{(doctor\_speciality, doctor\_speciality\_value), (doctor\_state, doctor\_state\_value), \dots\}$

“ $o_j$ ” field describes the object which’s the resource, the question of the access request. The object is defined by a list of attributes and their values.

$medical\_test = \{(medical\_test\_type, medical\_test\_value), (medical\_test\_status, medical\_test\_status\_value), \dots\}$

“ $a_i$ ” field describes the action which is a specific requested operation on the object “ $o$ ”. The action is defined by a list of attributes and their values.

$share = \{(share\_receiver, share\_receiver\_value), \dots\}$

Either for “ $s$ ” or “ $o$ ” or “ $a$ ”, the list of attributes gives a fine grained way for rules definition and expression. Finally, contextual constraints are useful to evaluate the context which a rule is applicable of a rule. We focus in temporal and special constraints.

$$\begin{aligned}
 Req_i(s_i \times a_i \times o_i \times [c_i]^*); i \in IN \\
 s_i &= \{(sa_j, sav_j); j \in IN\} \\
 o_i &= \{(oa_j, oav_j); j \in IN\} \\
 a_i &= \{(aa_j, aav_j); j \in IN\} \\
 c_i &= \{location, starttime, endtime\}
 \end{aligned}$$

**Formal definition (3):** *Extend an access control policy to specify privacy requirements*

In order to enforce the privacy compliance while expressing access control policies and rules, we extend previous definitions with privacy requirements. As a new combining algorithm we add “legal-overrides”

$$SP = (\{P_i, i \in IN\} \times CA)$$

$CA \in \{\text{deny-overrides, permit-overrides, first-applicable, only-one-applicable, legal-overrides}\}$

*legal-overrides:* In case of more than one rule that answers to a specific access request, we can override the legal rule in order to solve any issue about conflicts decisions. If more than one legal rule is applicable the first one of these legal rules will be chosen. If no rule at all is applicable, the deny response overrides as we are looking to avoid any risk about privacy preservation.

We extend definition (2) by adding a Rule type. Indeed, a policy is defined by its type ( $type_i$ ) and the associated rule ( $Rule_i$ ). The type of the rule will help us to classify rules. The classification could be helpful in order to set the priority of rules. In some cases, the priority is useful to resolve conflicts between rules. In our proposal, a rule could be “User Preference” rule or “Organization” rule or

“legal” rule. In this case of a “legal rule”, the rule must specify legal constraints  $[l_i]$ . Here constraints incorporate legislation specification such as the text law source location (the country, state, ...), the reference legislation (national law, international law, ...) and legal strength.

$$\begin{aligned}
 P &= (\{type_j \times Rule_j, j \in IN\} \times CA) \\
 type_j &\in \{legal, userPreference, organisation\} \\
 Rule_j &= Req_j \longrightarrow Resp_j \times [oblig_j]^*; j \in IN \\
 Req &= (s_j \times a_j \times o_j \times [c_j]^* \times [l_j]^*); j \in IN \\
 Resp &\in \{yes, no, undetermined\}
 \end{aligned}$$

In the same scope of privacy conformity, we improve the  $\prec s_i, a_i, o_i \succ$  attributes by explicitly provide privacy profile-based characterization (Fig. 7).

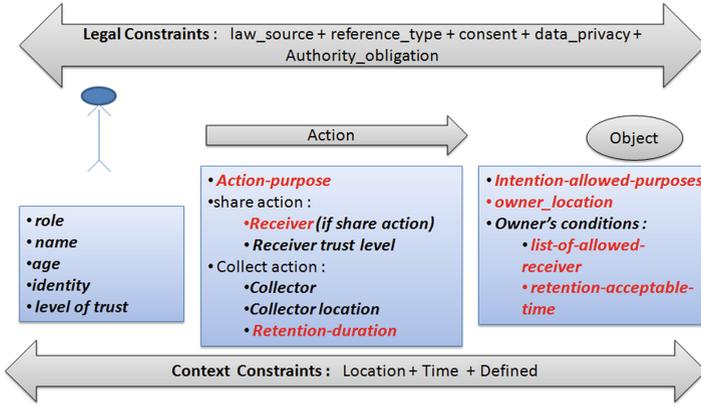


Fig. 7. Privacy characterizations in profile based access control

Then, in order to conclude about the privacy compliance of a request, we define some conditions to be respected. In the first condition, the “**action-purpose**” must belong to the set of “**intention-purpose-allowed**” allowed by the owner of the object. The second condition, if we deal with a disclose action with third party, the “**receiver**” must belong to the “**list-of-allowed-receiver**”. The third condition, the “**collection-retention-duration**” must respect the “**retention-acceptable-time**” by the data’s owner. The last condition is the “owner’s consent”.

### 3.4 Legal Constraints Reasoning

In this paragraph, some algorithms are proposed. The rule’s scheduling algorithm is described in (Fig. 8). It aims for the resolution of rules priority conflicts

```

// Sorting i Rules associated to a policy Pi
// Resolution of Possible Priority conflicts : make legal rule prior ones
For each Pi :
    tri(Rulej, legal-override, legal-strength)
        // Sorting Rules (1) : Place Rules with type Legal in first rules
        // Legal strength calculation for legal rules
        // Sorting Rules (2) : Tri legal rules according to legal-strength values

```

**Fig. 8.** Scheduling rules according to legal power

```

Read(access-Request)
    // Extract access characterisation
    // Extract privacy characterisation if possible
Rulej = Find-Suitable-Rule(access-Request)
    // Evaluate by by one rule ; the first applicable rule is returned
Make-Decision(access-Request, Rulej)
    // Check access entities characterisation
    // Check privacy conditions
    // Return evaluation of one request

```

**Fig. 9.** Check privacy compliance algorithm

$$legal_{strength} = \frac{(c_1 \times consent + c_2 \text{ dataPrivacy}_{Status} + c_3 \text{ Authority}_{obligation})}{3}$$

**Fig. 10.** Legal-strength calculation of one legal rule

by ordering security rules according to their type and their calculated power. So, “legal” rules type must be placed at the header of a rule. Then, this set of “legal” rules should be ordered based on their “legal strength” attribute. Another ongoing work (the associated paper is submitted), we proposed a formula calculating “legal-strength” of one “legal” rule based on legal conditions described in paragraph 3.3(Fig. 10). The “legal strength” calculation uses proportional coefficient defined by security administrator with the advice of a lawyer. The factors of evaluation are the number of legal conditions and their severity. For example, the authority obligation has the high severity. Another factor, the consent necessity, it depends on its conditions of specificity, format and destination criteria (see previous paragraph).

The decision making algorithm (Fig. 9) takes advantage of the previous Rules scheduling. It finds the first applicable rule, evaluates conditions with a special care for privacy conditions. So we get not only a security preserving access decision but also a privacy compliance decision.

## 4 Case Study

In order to provide the reader with a real situation of incorporating privacy in access control, we instantiate our generic model through the use of an example

<u><i>Security policies: Access control policies</i></u>	
<i>R1:</i>	The hospital save data patient tests results for a patient regular survey with a retention limited by 3 years.
<i>R2:</i>	Only doctors could modify patient health state interpretation.
<i>R3:</i>	If an information could threat national health security it should be "by force of law" disclosed to authority.
<i>R4:</i>	A practitioner could share a patient sensitive test medical result (for example: mammogram) if patient has provided informed consent for a specific purpose of processing and the processing purpose is compatible with the purpose contented for.
<i>R5:</i>	If UK medical data is to be processed by a medical professional for the purpose clinical research on breast cancer and the patient could be identified from the data. Then acquiring patient consent is necessary and consent must be an informed specific explicit consent.
<i>R6:</i>	If Italy, the consent can be given in a single, one-off statement (general consent). No provision for the need for explicit consent.
<i>R7:</i>	In France, express consent (written) is required for the processing of sensitive data.

**Fig. 11.** A set of informal access control policies in a hospital

**Table 1.** Scheduling rules according to a legal reasoning

Rule number	Rule type
R3	Legal
R4	Legal
R5	Legal
R6	Legal
R7	Legal
R1	Organization
R2	Organization

from the medical domain. We present the security policy of a hospital involved in a distributed search context. A context that involves multi-authority actors. The hospital in question is located in UK. It contributes to European breast cancer researches and shares medical results with some other European countries (Italy and France). They all work under the “Data Protection Directive” jurisdictions. In our case study, we imagine the hospital security policy and we describe it by a set of rules as specified in (Figs. 11 and 12). R1, R2 and R3 are organizational rules. R4, R5, R6 and R7 are legal based rules.

Based on previous algorithms for scheduling, the Table 1 shows the new ordered rules. Using our ontology we can make some inferences results on top of some requests of access. In this stage of work, we get limited inference possibilities regarding limits of owl in expressing rules. In a second stage of work, SWRL will be used to express rules in rich manner (Fig. 13).

Rule Number	Rule Type	Request fields			Response	Obligation
R1	Organization Rule	Subject	Role	Hospital administration	Permit	Retention limit is 3 years
		Object	Type	Patient data test		
		Action	Type	Save		
		General Constraints	Defined	Patient regular survey		
R2	Organization Rule	Subject	Role	Doctor	Permit	
		Object	Type	Patient data		
		Action	Type	Modify		
		General Constraints	Context	Health State interpretation		
R3	Legal Rule	Subject	Role	Law Authority	Permit	
		Object	Type	Patient data		
		Action	Type	Disclose		
		Legal Constraints	Law Source	Data Protection Directive		
			Reference Type	European Union Directive		
			Authority Obligation	yes		
General Constraints	Defined	Threat national health security				
R4	Legal Rule	Subject	Role	User	Permit	
		Object	Type	Patient data		
		Action	Intention Purpose	Same as Action purpose		
			Type	Share		
			Action purpose	Some purpose		
		Legal Constraints	Law Source	Data Protection Directive		
			Reference Type	European Union Directive		
			Consent	Yes		
			Consent specific	Specific		

Fig. 12. Primary formalization of informal access control rules (R1-R4)

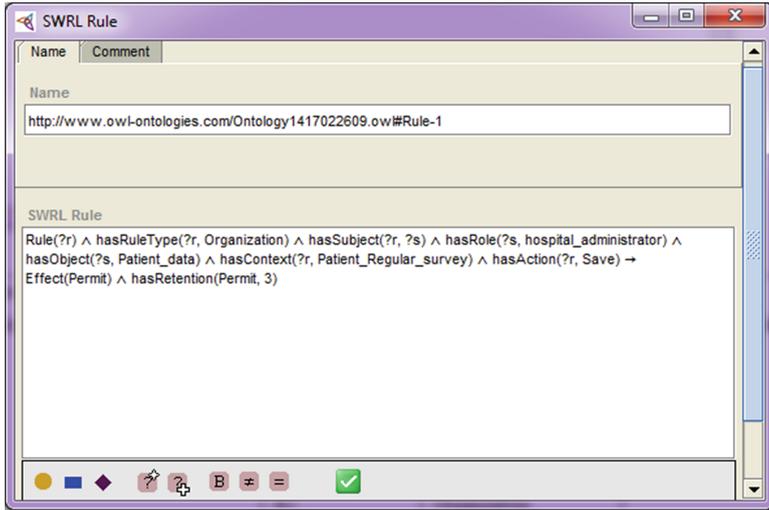


Fig. 13. Example of SWRL rule edition in Protégé (R1)

## 5 Related Work and Conclusion

In literature, many works have employed ontology while expressing privacy requirements in access control. In fact, Ontology-based Information modeling is considered as a power tool for logic-based inference and reasoning. One category of proposed ontologies [3, 4] have detailed in a clear semantic representation

privacy rules. But it didn't deal with legislation that are the primary source of mentioned privacy requirements. Also it lacks answers on how to deal with ordering rules and possible conflicts resolution. Other category of proposed ontologies [2, 27] focus on access control requirement representation.

These works demonstrate how ontology could be a useful tool for interoperability handling in open environments. However in this position, privacy preserving was not subject to checking while making access control decisions.

In this paper, we suggest a semantic formalization of access controls that ensures compliance with privacy requirements that are imposed by legislation. In order to achieve our goal, we take advantage of a semantic web standard for ontology representation this is because; ontologies could provide simple key tools to govern policy information heterogeneity over different domains in complex distributed environments. Moreover, we propose to incorporate references to text law and the legislative enforcement strength while expressing access control policies. Besides, it could be useful for some cases to resolve conflicts between access control rules at execution time.

For future work, we are working on extending the XACML standard architecture for access control. For this purpose we aim to build an ontology driven access control architecture. This could be presented as a distributed architecture with an added semantic layer which allows the integration of fine grained privacy requirements. Besides, we are looking to put together an ontology reasoning engine for legal strength estimation. We aim, in this context, to provide an engine that calculates a score of each legal privacy policy. This score is evaluated according to the law reference and an assessment of the weighting of the referred text compared to the reference law enforced by other intervening access control rules or policies. In addition, we are planning to work on improving rules inference engines by extending or proposing new inference systems ensuring conflict detection (e.g. duplication and contradiction) between rules.

## References

1. Damiani, E., Samarati, S.: New paradigms for access control in open environments. In: Proceedings of the Fifth IEEE International Symposium Signal Processing and Information Technology, pp. 540–545 (2005)
2. Reul, Q., Meersman, R.: Ontology-based access control policy interoperability. In: STARLab (2013)
3. Zhang, N.J., Todd, C.: A privacy agent in context-aware ubiquitous computing environments. In: Leitold, H., Markatos, E.P. (eds.) CMS 2006. LNCS, vol. 4237, pp. 196–205. Springer, Heidelberg (2006)
4. Garcia, F.: Towards a base ontology for privacy protection in service-oriented architecture. In: IEEE International Conference on Service-Oriented Computing and Applications (SOCA) (2009)
5. Gruber, T.R.: A translation approach to portable ontology specifications. *Knowl. Acquisition* **5**(2), 199–220 (1993)
6. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum. Comput. Stud.* **43**(5–6), 907–928 (1995)

7. Spyns, P., Meersman, R.: An ontology engineering methodology for DOGMA. *Appl. Ontology* **3**(1–2), 13–39 (2008)
8. Byun, J., Li, N.: Purpose based access control of complex data for privacy protection. In: *Proceedings of the Tenth ACM Symposium on Access Control Models and Technologies*. ACM New York (2005)
9. Covington, M.J., Sastry, M.R.: A contextual attribute-based access control model. In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM 2006 Workshops*. LNCS, vol. 4278, pp. 1996–2006. Springer, Heidelberg (2006)
10. Gilles, N., Kamel, M.: Ontology learning by analyzing XML document structure and content. In: *Proceedings of the International Conference on Knowledge Engineering and Ontology Development KEOD Portugal* (2009)
11. Kamel, M., Rothenburger, B.: Eliciting hierarchical structures from enumerative structures for ontology learning. In: *Proceedings of the 6th International Conference on Knowledge Capture K-CAP* (2011)
12. Noy, N.F., McGuinness, D.: An ontology development 101: a guide to creating your first ontology. *Stanford knowledge systems laboratory Technical report KSL-01-05 and stanford medical informatics Technical report SMI-2001-0880* (2001)
13. Ben Abacha, A., Da Silveira, M., Pruski, C.: Medical ontology validation through question answering. In: Peek, N., Marín Morales, R., Peleg, M. (eds.) *AIME 2013*. LNCS, vol. 7885, pp. 196–205. Springer, Heidelberg (2013)
14. Noy, N.F., Musen, M.A.: The protégé OWL plugin: an open development environment for semantic web applications. In: McIlraith, S.A., Plexousakis, D., Harmelen, F. (eds.) *ISWC 2004*. LNCS, vol. 3298, pp. 229–243. Springer, Heidelberg (2004)
15. Protege. <http://protege.stanford.edu>
16. SWRL. <http://www.w3.org/Submission/SWRL/>
17. OWL. <http://www.w3.org/TR/owl-guide/>
18. OECD Privacy. <http://www.ncbi.nlm.nih.gov>
19. EC: Data Protection in the European Union. European Commission (2010)
20. Boussi, H.: Ontology based privacy compliance for health data disclosure in Europe. A thesis report (2010)
21. Caralt, N.: Modelling legal knowledge through ontologies. A thesis report (2008)
22. Horrocks, I.: OWL: a description logic based ontology language. In: van Beek, P. (ed.) *CP 2005*. LNCS, vol. 3709, pp. 5–8. Springer, Heidelberg (2005)
23. Zhu, J., Smari, W.W.: Attribute based access control and security for collaboration environments. In: *Aerospace and Electronics Conference* (2008)
24. Sandhu, R., Park, J.: Usage control: a vision for next generation access control. In: Gorodetsky, V., Popyack, L.J., Skormin, V.A. (eds.) *MMM-ACNS 2003*. LNCS, vol. 2776, pp. 17–31. Springer, Heidelberg (2003)
25. Oasis Web Site (2013). <http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html>
26. Anderson, A.H: A Comparison of Two Privacy Policy Languages: EPAL and XACML. *GSun Microsystems Labs Technical report* (2005)
27. Özgü, C.A.N., Bursa, O., Ünalir, M.O.: Personalizable ontology-based access control. *Gazi Univ. J. Sci.* **23**(4), 465–474 (2010)



<http://www.springer.com/978-3-319-31810-3>

Risks and Security of Internet and Systems  
10th International Conference, CRiSIS 2015, Mytilene,  
Lesbos Island, Greece, July 20-22, 2015, Revised  
Selected Papers  
Lambrinoudakis, C.; Gabillon, A. (Eds.)  
2016, X, 307 p. 72 illus., Softcover  
ISBN: 978-3-319-31810-3