A SWRL Bridge to XACML for Clouds Privacy Compliant Policies

Hanene Boussi Rahmouni[1,3], Marco Casassa Mont[2], Kamran Munir[1] and Tony Solomonides[1,4]
[1]*Department of Computer Science and Creative Technologies, Faculty of Environment and Technology, University of the West of England, Coldharbour Lane, Bristol, BS16 1QY, U.K.*
[2]*Hewlett-Packard Labs, Cloud & Security Lab, Bristol, U.K.*
[3]*Higher School of Communication, University of Carthage, Tunis, Tunisia*
[4]*Center for Clinical and Research Informatics, North Shore University Health System, Illinois, U.S.A.*

Keywords: Privacy Policies, OWL, SWRL, XACML, Cloud.

Abstract: The management of privacy and personal information within multi-cultural domain such as clouds and other universal collaborative systems requires intrinsic compliance-checking and assurance modules in order to increase social trust and acceptance. Focusing mainly on medical domains, this issue is particularly important due to the sensitivity of health related data in international data protection law. The use of ontologies and semantic technologies can provide relatively easy interpretation of legislation at run time, and can allow the logging of data access events to serve for future audits. However, the enforcement of semantic web rules (SWRL rules) on complex and heterogeneous architectures is expensive and might present runtime overheads. We believe a mapping of our semantic web privacy policies to a standard access control language such as XACML would be a useful alternative. A translation to XACML, would allow the integration of these policies with existing security and privacy policies being adopted on clouds environments. This paper describes a mathematical formalism for mapping SWRL (Semantic Web Rule Language) privacy rules to XACML policies and also explains the underline implementation requirements of this formalism.

1 INTRODUCTION

The protection of patients' privacy in a pan-European cloud infrastructure is challenging and requires combined solutions from legislation, organisational and social frameworks. In this regards, a European public cloud infrastructure is still a challenging goal to attend (Brandic et al., 2010), but necessary for those nations wishing to collaborate for the advancement of medical research and public health. This challenge arises primarily due to the lack of harmonisation in legal frameworks governing privacy and data protection in Europe, not least the European Data Protection directive 95/46/EC (EC Directive, 1995), (McCullagh, 2006), (Beyleveld et al., 2004). For example, consent is not handled in the same way in Italy as in the UK. In Italy, consent could be provided for a broad purpose of data processing; whereas in the UK, obtaining a specific consent is a legal obligation (Iversen et al., 2006), (Italian Personal Data Protection Code, 2003). On top of this, there are significant conceptual and technical issues in-particular when expressing, interpreting and deriving operational consequences out of high-level policies. Finally, despite the attention that has been paid to security concerns for public and private clouds; such as infrastructure integrity and access control (typically authentication and authorization), this does not naturally extend to cover privacy concerns (often requiring context and purpose specification). Although they are newly emerging paradigms, clouds are very similar in many aspects to other distributed computing environments. Particularly, clouds are similar to large-scale systems that are based on virtualised technologies such as Grid systems (OCSI, 2010). These systems have high capabilities for sharing data and resources through the Internet. However they often fall short of providing measurable proof of compliance, which is required throughout the complete data sharing process. This is different than the case of traditional centralised systems, where the data security focus was directed only towards data access transactions. As such, it is an on-going challenge to search for ways to narrow the gaps between the various *legal*,

Boussi Rahmouni H., Casassa Mont M., Munir K. and Solomonides T..
A SWRL Bridge to XACML for Clouds Privacy Compliant Policies.
DOI: 10.5220/0004853900270037
In *Proceedings of the 4th International Conference on Cloud Computing and Services Science* (CLOSER-2014), pages 27-37
ISBN: 978-989-758-019-2

technical, *social* and *organizational aspects* of the problem.

The approach presented in this paper is an attempt to show that the use of Semantic Web technologies (Horrocks et al., 2004) can allow both the specification and enforcement of privacy requirements that traditional access control languages and mechanisms cannot achieve. We start from the high-level regulations that govern privacy and data protection in Europe and we progress towards the integration of privacy constraints interpreted from them within access controls specifications. For this matter, policies' decisions cannot be deduced from data identifiers and access control conditions that are evaluated against their attributes' values. Instead, the evaluation of privacy policies requires more information about resources; and hence, we face the need to record metadata about the protected resources in a computational infrastructure. We believe, the existing access control solutions for the cloud need to evolve in order to allow for such integration and in order to enable enforcement of the full range of *privacy* constraints.

Although semantic based languages can adequately capture and conceptually specify the *contexts*, *facts* and *rules* necessary for reasoning about data manipulation obligations, it is rather not suitable for implementation in a cloud context. This is due to the necessity for answering two major clouds requirements namely *performance* and *standardisation*. In order to enable better interoperability, while exchanging data in the cloud, it is important to use standard data management languages and services. This includes both standard access control and security languages (OCSI, 2006). Moreover, the use of semantic access control languages requires customised enforcement architectures that are different from the ones adopted on the cloud infrastructure and that are designed to enforce policies specified in a standard format. A similar change might be very expensive from the point of view of clouds services and infrastructure providers. In order to be easily enforced at the cloud's system-level, we suggest that the presented policies should eventually be specified in a way that conforms to a widely adopted policy language or standard. In particular, a standard that has proven efficiency in the enforcement of privacy policies. Our choice is the eXtensible Access Control Mark-up Language (XACML) (OASIS XACML, 2005). It is worth mentioning; and in order to eliminate confusion, that in the context of this work, we do not claim that XACML can handle privacy constraints in exactly the same way as it handles security constraints. The limitations of XACML, both as a policy language and as an enforcement mechanism, have been detailed in the literature (Casassa et al., 2007), (Sommer et al., 2008), (Casassa, 2010). Also additional limitations are presented in Section 3 of this paper. In this work, we seek to overcome some of these limitations. For a note to the readers, additional effort was also made in later version of XACML (OASIS XACML, 2013).

The remaining paper is organised as follows: Section 2 starts by clarifying the theory presented in this paper in comparison and continuation to the allied work that we have been doing previously in this domain. This Section also clarifies the major contributions presented in this paper. Section 3 presents a synopsis of the main technologies on which we have based our privacy specification and enforcement approach, which are presented in later Sections 4, 5 and 6. In particular, Section 4 discusses the SWRL-based privacy policies specification and Section 5 shows how they could be rewritten in a syntax conforming to the XACML standard. In Section 6, a formalism for mapping SWRL privacy rules into XACML access controls is presented. This is followed by the requirements and recommendations for implementing the projected formalism in Section 7; and finally, the conclusion and relevant future orientations are presented in Section 8.

2 PAPER CONTRIBUTION AND RELATION TO PREVIOUS WORK

In (Rahmouni et al., 2010) and (Rahmouni et al., 2011), we have described how Semantic Web technologies have been used to classify the resources that we would like to protect. At that stage the resources were specified using the metadata captured within an ontology. We have also shown in this existing work that how different scenarios of data/resource sharing have been modelled within the same ontology. In this paper, we describe extensions to the previous model (with necessary metadata added) and extend the data sharing scenarios to include privacy policy contexts. We then show how this allows the specification and editing of privacy and access control policies in terms of existing concepts within the ontology. There is research reported in the literature; such as (Muppavarapu and Chung, 2008), (Gowadia et al., 2008) and (Matteucci

et al., 2010), that has looked at the use of ontologies and Semantic Web technology in order to allow a better specification and enforcement of security and authorisation policies. Among these, only the "Consequence" project has looked at an approach that integrates requirements from high-level policies through the means of controlled natural language (Matteucci et al., 2010). This approach translates high-level policies extracted from data sharing agreements into a natural language-like formalism in order to allow enforceability. This work did not stop at the *control of access to data*, but has rather focussed on ways of controlling any type of data usage even after the data were shared with a party belonging to an external domain. This functionality is worth further consideration and is discussed in the future work section of this paper. In comparison, the actual status of our approach allows the disclosure of data handling policies to external parties receiving personal data, but does not enforce these policies within the receiver's domain. However, the work in (Matteucci et al., 2010) included little effort to integrate within access control policies, privacy requirements that are interpreted from primary legislation (text law). It made rather a focus only on traditional security services such as authorisation and the trust aspect of it. Hence this approach couldn't fit in a solution aiming for the big picture of regulatory compliance. This is because usually traditional security requirements covers only a very specific subset of jurisdictional requirements that are not general enough to cover any case of data sharing that might arise in the future.

3 SWRL AND XACML

In this section, an overview of the semantic web rule language (SWRL) and the extensible access control markup language XACML is presented. In this regard, an analysis of their expressiveness capabilities and utility for enforcing privacy policies in a cloud environment is also elaborated.

SWRL, the Semantic Web Rule Language (SWRL) (Horrocks et al., 2004) is based on a combination of the OWL-DL (Matteucci et al., 2010) and some sublanguages of the Rule Mark-up Language (RuleML) (Boley et al., 2010). SWRL includes a high-level abstract syntax for Horn-like rules in both the OWL-DL and OWL-Lite sublanguages of OWL (Bechhofer et al., 2004). The proposal extends the set of OWL axioms to include Horn-like rules. It thus enables the rules to be combined with an OWL knowledge base. Some model-theoretic semantics are given to provide the formal meaning for OWL ontologies, including rules written in an abstract syntax. With the combination of an XML syntax based on RuleML, the OWL XML Presentation Syntax and an RDF (Bechhofer et al., 2004) concrete syntax based on the OWL RDF/XML exchange syntax, SWRL presents an illustration of the extension of description logic into defeasible description logic (Wang et al., 2004). This makes it a promising technology for the modelling of regulations.

The proposed rules are of the form of an implication between an *antecedent (body)* and a *consequent (head)*. The intended meaning can be read as: *whenever the conditions specified in the antecedent hold, then the conditions specified in the consequent must also hold.* Both the *antecedent (body)* and *consequent (head)* consist of *zero* or *more* atoms. An *empty antecedent* is treated as *trivially true* (i.e. satisfied by every interpretation), so the consequent must also be satisfied by every interpretation; an *empty consequent* is treated as *trivially false* (i.e., not satisfied by any interpretation), so the antecedent must also not be satisfied by any interpretation. Multiple *atoms* are treated as a *conjunction*. Note that rules with *conjunctive* consequents could easily be transformed into multiple rules, each with an *atomic consequent* (Gruber, 1995). Atoms in these rules can be of the form *C(x), P(x,y), sameAs(x,y)* or *differentFrom(x,y),* where *C* is an OWL description class, *P* is an OWL property, and *x,y* are either variables, OWL individuals or OWL data values. It is easy to see that OWL DL becomes undecidable when extended in this way as rules can be used to simulate role value maps (Gruber, 1995).

XACML (OASIS XACML, 2005) is an XML specification and syntax for expressing policies controlling the access to information through the Internet. It provides the enterprises with a flexible and structured way of managing access to resources. The specification language is based on a subject-target-action-condition policy syntax specified in an XML document. As specified in the Fig. 1 (OASIS XACML, 2005) a *Policy* is composed of a *Target*, which identifies the set of capabilities that the requestor must expose along with a set of rules varying from one to many. Every Rule contains the specific facts needed for the access control decision-making. It also has an evaluation Effect, which can be either *Permit* or *Deny*. At policy evaluation time a policy combining algorithm is used to deal with *(permit/deny)* conflicts which might arise in the rule decisions. A *Target* is composed of four sub-

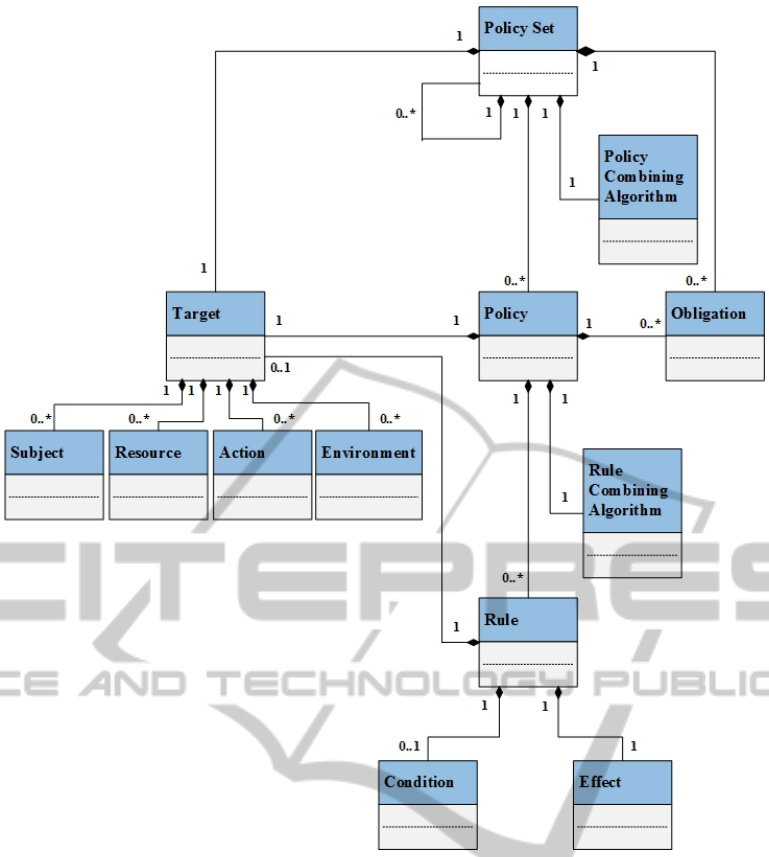

Figure 1: XACMLv2's data flow model.

elements: *Subjects*, *Actions*, *Resources*, and Environments. Beyond what is described in the Fig. 1, each target category is composed of a set of target elements, each of which contains an attribute identifier, a value and a matching function. Such information is used to check whether the policy is applicable to a given request. This could be specified in the condition section of a rule.

In most of the cases the language defines controls as a collection of attributes relevant to a principle. It includes both conditional authorisation policies and other policies to specify post conditions such as notifications to data subject. Like other policy languages that are based on XML, XACML lacks the required semantics to handle heterogeneity and permits interoperability, especially when managing data access within environments that involve multiple organisations. The different data access requests coming from users in different organisations might refer to the same data item with different naming. Additional semantics are needed in order to allow semantic alignment to the different terms used to describe the same data item (Muppavarapu and Chung, 2008). Moreover dealing with dynamic attributes such as the user's age or hierarchical attributes for example the user's role requires some additional semantics and integrated reasoning (Demchenko et al., 2008), (Priebe et al., 2006), (Damiani et al., 2004).

4 A SWRL-BASED PRIVACY POLICY SPECIFICATION

We have examined the legal privacy rules and obligation dictated in many jurisdictional texts and we have noted that the rules are specified according to a specific vocabulary describing many conceptual entities. These entities are usually associated together in different combinations in order to build generic rules that could be modelled in the form of *if-then* rule template. Following this assumption and similarly to the work done in (Powers et al., 2004) that simplifies policies dictated by the Ontario's Freedom of Information and Protection of Privacy Act (FIPAA) (Ontario, 2008), we have expressed the policies specified in European data protection text

law in a more simplified way using the different concepts that constitute its vocabulary. These concepts were captured in an OWL ontology that we have described in previous work (Rahmouni et al., 2010) and (Rahmouni et al., 2011). The policies were then matched to the rule template.

Privacy-Rule-Template:

If

[Context] and [Condition on User], [Condition on data], [Condition on Purpose], [Condition on Other] (including checking for privacy requirements)

Then

Allow [action] and Impose [Obligation]

The rule *privacy-rule-template* could be adapted and specialised, according to the context of application, in order to represent privacy requirements in a case based manner. On this basis, we have rewritten privacy policies interpreted from text law as SWRL rules using OWL classes and properties specified in our privacy ontology. The rule conforms syntactically to the SWRL human readable syntax:

Antecedent Clause *implies* Consequence Clause

Or, in a different notation:

`Antecedent → Consequent`

Adapting the rule to an access control policy format, it must conform to the following template:

```
Rule := Target ∧ Conditions → Effect ∧
        Obligations
```

Here we explain our SWRL privacy policies specification through a concrete rule example and a cloud data sharing scenario:

Example: Purpose Compatibility Rule

In order to clearly explain our approach we start by specifying an example of high-level policy extracted from European privacy legislation. The policy is further taken trough series of transformations towards an operational status in the format of XACML syntax. In this example, we show how we model the privacy policy stating (Iversen et al., 2006) that:

"A user may access a patient mammogram for a stated purpose provided that the patient has given informed consent for a specific processing purpose and the stated processing purpose is compatible with the purpose consented for".

We present below the application of the generic template of SWRL privacy rules to this rule example. For this we adopt a human readable SWRL syntax.

We denote by `(R, T, Con, E and Ob)` respectively the Rule elements `(Rule, Target, Conditions, Effect and Obligations)` described in the abstract syntax of privacy rules given above. The rule template is therefore rewritten as follows:

`R := T ∧ Con → E ∧ Ob`

In order to implement our rule example, we need to apply it to a concrete data sharing scenario. For this we present the example of data sharing in the cloud described in Fig. 2:

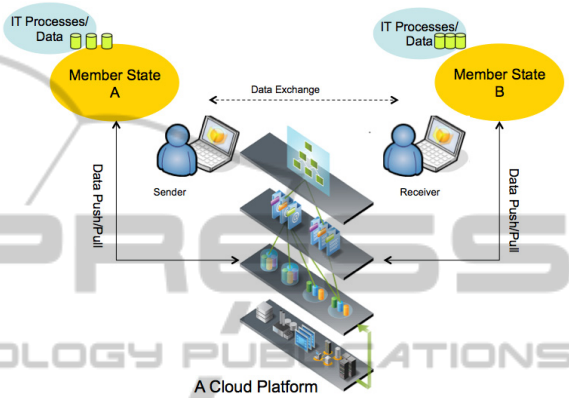

Figure 2: A cloud data sharing scenario.

The scenario we have chosen describes a case of data sharing in the *health* domain. We assume that two medical doctors belonging each to a different hospital in different European member states for example UK and Italy form the two data sharing parties. To be more precise, one of the medical doctors would like to get a second opinion on a patient's *Mammogram*.

The data will be exchanged on a cloud platform and it is required that the cloud security services could identify the right policy to apply in order to allow the sharing of the data, but in a lawful way. Since we are looking at a pan-European context, it wouldn't be always the case that the data processing law is interpreted and implemented in one member state in exactly the same way as in another. Stating as an example, when processing health data for the purpose of medical research, the patient consent must be a *specific consent* when referring to the law in the UK or France. However consent could be *broad* or *general consent* when referring to an Italian law.

For this more context information should be provided in the privacy rules specification for the cloud security processes in order to be able to make the right decision. It is therefore essential to indicate in the rule implementation the sender and receiver's locations and the member state from which the shared data comes. An instantiation of R in the context of the

rule example and the cloud data sharing scenario is then interpreted in the Fig. 3.

```
T:
          hasSender(?x, ?s)
       ∧  hasReceiver(?x, ?r)
       ∧  hasPurpose(?x, ?p)
Con:
       ∧  locatedIn(?s, UK)
       ∧  locatedIn(?r, Italy)
       ∧  concerning(?x, ?m)
       ∧  belongsTo(?m, UK)
       ∧  isForPatient(?m, ?pt)
       ∧  provided(?pt, InformedConsent)
       ∧  hasCollectionPurpose(?m,
          CollectionPurpose)
       ∧  compatibleWith(?p,
          CollectionPurpose)
E:    hasSharingDecision(?x, allow)
Ob:      hasObligation(?x,
         attachSecondaryUsePolicy)
```

Figure 3: Instantiation of the privacy rule template.

In this SWRL rule example OWL properties and classes were used to describe the different elements of the privacy rule target *T,* for example *hasSender* is an owl object property specifying the sender *s* involved in the data sharing ?*x*. Other properties are also used to declare *T* including *hasReceiver*, *hasPurpose* for specifying the receiver of the data and the purpose of sharing respectively. The OWL property *concerning* is used to capture the resource being shared. Since the scenario involves the sharing of patient *?pt* mammograms we have denoted the shared resource/object as *?m*.

The second part of the rule antecedent are the conditions section and it shows constraints the target elements should satisfy in order to infer the effect and obligations shown in the rule consequent section.

5 MAPPING AN ACCESS CONTROL SWRL RULE TO AN XACML CONFORMING SWRL RULE

For easy mapping to an XACML rule, the SWRL rule has to be specified in terms of attributes of only the generic entities that constitute an XACML Rule Target (see above) and other elements that are used to specify the general policy that the rule in question belongs to, e.g. the purpose of processing. In this regard, the OWL property:

`provided(Patient, Informed Consent)`

is a property of the patient whose data is to be shared and indicates that the patient has provided informed consent. The patient or the data subject is not one of the XACML "Rule Target" components; therefore, we express the same condition in terms of property of the class "O" (the resource or object. In our case it is the data the subject is requesting access to). The result is presented in Table 1:

Note: we do not need to translate provided (?pt, InformedConsent) as we are not keeping constraints about patients in the XACML version of the rule. For example, in the XACML conforming SWRL Rule, the consent is an attribute of the object and not of the patient any more.

The rule described above is an extension or privacy aware version of traditional access control rules that pays no significant attention to privacy constraints and obligations. If specified in SWRL syntax, an

Table 1: Mapping from SWRL rule to XACML conforming SWRL rule.

Initial SWRL Privacy-Aware access control rule	XACML Conforming SWRL Privacy-Aware access control rule
dataSharing(?x)	hasRuleContext(?r, ?rc)
∧ hasAction(?x, ?ac)	∧ hasContextAction(?rc, ?ac)
∧ hasSender(?x, ?s)	∧ hasContextSubject(?rc, ?s)
∧ hasReceiver(?x, ?r)	∧ hasReceiver(?ac, ?rec)
∧ concerning(?x, ?m)	∧ hasContextObject(?rc, ?o)
∧ hasPurpose(?x, ?p)	∧ hasContextPurpose(?rc, ?p)
∧ action(?ac, send)	∧ action(?ac, send)
∧ locatedIn(?s, UK)	∧ locatedIn(?s, UK)
∧ belongsTo(?m, UK)	∧ consent(?o, true)
	∧ hasConsentType(?o, InformedConsent)
∧ compatibleWith(?p,CollectionPurpose)	∧ compatibleWith(?p, ?cp)
Rule Implication	
hasSharingDecision(?x, allow)	hasRuleEffect(?r, allow)
∧ hasObligation(?x, attachSecondaryUsePolicy)	∧ hasObligation(?r, attachSecondaryUsePolicy)

example of this traditional access control rule would look as shown in Fig. 4.

```
hasRuleContext(?r, ?rc)
∧     hasContextSubject(?rc, ?s)
∧     hasContextObject(?rc, ?o)
∧     hasContextAction(?rc, send)
∧     hasContextPurpose(?rc, ?p)
∧     hasRole(?s,doctor)
∧     isForPatient(?o,?pt)
∧     isDoctorOf(?s,?pt)
→     hasRuleEffect(?r, allow)
```

Figure 4: Instantiation of the privacy rule template.

The only constraints the rule above tests for before allowing the disclosure of the data is the role of the *subject* or *requestor*. In this case, the role of the subject must be a medical doctor of the patient whose data is to be disclosed.

XACML was designed to notate access control policies and to provide a reference framework for their enforcement. Its major focus is on security policies, although privacy is mentioned in the specification of version 2.0 (OASIS XACML, 2005). It is verbose and complex and still lacks expressiveness. The XACML version 3.0 however, seems to provide a better privacy specification profile (OASIS XACML, 2013). We have also noticed that some examples included in the XACML privacy profile, which were supposed to specify a policy compliant with the "specific and compatible purpose" privacy principle, in fact test for equality or matching of purposes rather than compatibility of purposes. We believe this is due to the language's lack of semantics and reasoning ability with regards to privacy constraints on protected data. If this lack is not addressed, a straightforward mapping from SWRL policies to XACML will not be possible. From the examples of the SWRL access control and privacy rules presented earlier in this paper, we conclude that privacy obligations should be specified for each rule as they are matched according to the data sharing context that we declare to be unique for each rule. This is different from the way obligations are specified in XACML. Obligations in XACML are related to a policy and not to the individual rules that a policy is made up of. We have resolved this problem by allowing each policy to include only one rule and its applicable obligations. Indeed, this decision was already implicit at the time we designed our SWRL privacy aware access control policies. For it, we decided to include one rule per policy. In fact, dealing with more than one privacy obligation at once might require a large amount of contextual information. Therefore, the equivalent SWRL rule would become too long and less readable.

6 MAPPING AN XACML CONFORMING SWRL RULE TO AN XACML POLICY

In this section, we present an attempt to formalise a mapping of a SWRL rule to an XACML policy.

There is some existing work that has looked at formalisms of XACML with many purposes in mind such as in (Kolovski et al., 2006), (Kolosvki et al., 2008), (Kolovski and Hendler, 2008), (Masi et al., 2012) and (Jeremy et al., 2007). In particular, the work presented in (Kolovski et al., 2006) and (Kolosvki et al., 2008) has started from a BNF representation of an XACML rule and has produced a DL formalism that allows the mapping of an XACML rule to DL syntax. Our approach takes into consideration the syntactic difference between DL and SWRL. SWRL is an extension of OWL-DL that can be mapped to DL syntax. It has inherited *Horn-like* propositional logic syntax from RuleML and this characteristic would influence the deviation from a DL formalism provided in (Kolovski et al., 2006) and (Kolosvki et al., 2008). The mapping process has already started from the previous section when we have transformed our SWRL access control rule into an XACML conforming representation. This was done by translating all the properties occurring in the antecedent and consequent to properties applied only on concepts that could be identified in the set of entities that occur in the XACML language model. After the transformation, we suggest that our SWRL access control rules can be generalised under the following formalism.

Formalism1:

```
Rule  := Target ∧ Conditions
         → Effect ∧ Obligations
R := Tgt ∧ Con → Eft ∧ Ob
```

We denote by:

- **R:** an OWL concept representing an access control rule.
- **Tgt:** the target of a rule R that usually constitutes of the elements Subject, Object, Action and Purpose.
- **Con:** the constraints to be imposed on the different elements of a target and that should be satisfied in order for the decisions specified in the consequence clause to be satisfied.
- **Eft:** the effect of a rule R that could be a Permit

or Deny

- **Ob:** the set of obligations that could be associated with the rule R

Formalism 1 may be mapped to Formalism 2 as described below:

Formalism 2:

$$\text{Rule}\left(\text{R}\right)\wedge[\overset{3}{\underset{i=0}{\wedge}}\text{Pd}_i\left(\text{R},\ \text{pd}_i\left(\text{R}\right)\right)]\wedge[\underset{i<3,k<n}{\overset{m}{\wedge}}\text{Pc}_k\left(\text{C}_i,pc_k\left(\text{C}_i\right)\right)]\rightarrow \text{Eft}\left(\text{R},\text{e}\right)\wedge[\underset{j=0}{\overset{m}{\wedge}}\text{Ob}_j\left(\text{R},ob_j\left(R\right)\right)]$$

Where i, j and k are natural numbers ranging, respectively, over the number of rule target elements (0..3), the number of properties in our ontology (0..n), and the number of obligations that would be associated with the rule R (0..m), and where:

- $\wedge$ with limits is the symbol for multiple conjunctions;
- **Pd** denotes an OWL property used for declaration of the target elements of a rule R;
- **Pc** denotes an OWL property used for specifying constraints on the elements constituting a target of a rule;
- **C** represents a given class of our ontology with C0, C1, C2 and C3 representing respectively the entities constituting the elements of a target of a rule in the order: Subject, Object, Action and Purpose;
- **Eft** is an OWL property specifying the effect of a rule R, its value is a literal e where e belongs to {permit, deny}.

Table 2 provides a one to one mapping of the entities

Table 2: Logical formalism of SWRL access control rules.

Entity	SWRL formalism
SWRL Rule: SR	$\text{Rule(R)}\wedge[\overset{3}{\underset{i=0}{\wedge}}\text{Pd}_i\left(\text{R},\ \text{pd}_i\left(\text{R}\right)\right)]\wedge[\underset{i<3,j<n}{\wedge}\text{Pc}_k\left(\text{C}_i,pc_k\left(\text{C}_i\right)\right)]\rightarrow\text{Eft}\left(\text{R},\text{e}\right)\wedge[\overset{m}{\underset{j=0}{\wedge}}\text{Ob}_j\left(\text{R},ob_j\left(R\right)\right)]$
Effect	`Eft(R, e) e ::= Permit | Deny`
Target	`Subject(R,Sub)∧ Object(R,Obj)∧ Action(R,Act)∧ Purpose(R,Pur)` $::=\overset{3}{\underset{i=0}{\wedge}}\text{Pd}_i\left(\text{R},\ \text{pd}_i\left(\text{R}\right)\right)$
Conditions	`Conditions(Sub) ∧ Conditions(Act) ∧ Conditions(Res)∧ Conditions(Pur)::=` $\underset{i<3,j<n}{\wedge}\text{Pc}_k\left(\text{C}_i,pc_k\left(\text{C}_i\right)\right)$
Obligations	$\pi\text{Ob}::=\overset{m}{\underset{j=0}{\wedge}}\text{Ob}_j\left(\text{R},ob_j\left(R\right)\right)$

Table 3: SWRL to XACML mappings.

Entity	SWRL formalism	XACML formalism
Rule	`Formalism 2`	`R ::= (Rule Tgt Eft)`
Effect	`Eft(R, e) e ::= Permit | Deny`	`Eft ::= Permit | Deny`
Target	`Subject(R,Sub)∧ Object(R,Obj) ∧ Action(R, Act) ∧ Purpose (R, Purpose)` $::=\overset{3}{\underset{i=0}{\wedge}}\text{Pd}_i\left(\text{R},\ \text{pd}_i\left(\text{R}\right)\right)$	`Tgt ::= ((Sub) (Act)(Res) (Pur))`
Condition Clause	`πConditionsπ Sub) ∧ Conditionsπ(Act) ∧ Conditions∧π(Res) ∧ Conditions∧π(Pur)` $::=\underset{i<3,j<n}{\wedge}\text{Pc}_k\left(\text{C}_i,pc_k\left(\text{C}_i\right)\right)$	`Each Pc(C, Pc(C)) := Sub |Act | Res| Pur ::= Any | Fn Fn::= AV | Fn \ Fn | Fn [Fn | ¬ Fn AV::=(attr-id attr-val) attr-id attr-value`
Obligations	$\pi\text{Ob}::=\overset{m}{\underset{j=0}{\wedge}}\text{Ob}_j\left(\text{R},ob_j\left(R\right)\right)$	`Each Ob(R, ob(R))::= AV`

constituting an XACML rule and Formalism 2.

In order to be able to translate our SWRL rules to XACML rules we suggest allowing a one to one mapping between our SWRL Formalism 2 and the BNF formalism of an XACML rule provided in (Kolovski et al., 2006) and (Kolosvki et al., 2008). To achieve this, we have extended the XACML BNF notation with the purpose element of a rule target and the rule obligations clause. The mapping is described in Table 3.

Based on the above *one to one* mapping, we mapped the purpose compatibility SWRL rule produced earlier to the XACML Rule presented in Fig. 5. In this rule, we have chosen to name the *sender* and *receiver* specified previously in the cloud scenario (Section 4) as *Dr_House* and *Dr_Casa* respectively. Other context variables describing the objet to be shared and the sharing purpose were also replaced with concrete values. We have chosen *M1* to indicate the mammogram being sent by *Dr_House* and the sending purpose were specified as *SecondOpinionOnTreatment.*

```
<Rule RuleId = "1" Effect="Permit">
<Target>
<Subjects>< Attribute AttributeId="Subject-Id" DataType= "String">
<AttributeValue> Dr_House </attributeValue>
< Attribute AttributeId = "Location" DataType= "String">
<AttributeValue> UK </attributeValue>
< Attribute AttributeId = "Receiver-Id" DataType= "String">
<AttributeValue> Dr_Casa</attributeValue>
< Attribute AttributeId = "Receiver-Location" DataType= "String">
<AttributeValue> Italy </attributeValue>
< Attribute AttributeId = "Role" DataType= "String">
<AttributeValue> Doctor </attributeValue>
</Subjects>
<Resources>< Attribute AttributeId = "ResourceId" DataType= "String">
<AttributeValue> M1</AttributeValue>
< Attribute AttributeId = "Consent" DataType= "Boolean">
<AttributeValue> true</AttributeValue>
</Resources>
<Action> >< Attribute AttributeId = "Action-Id" DataType= "String">
<AttributeValue>send</AttributeValue></Action>
<Purpose>
<Attribute AttributeId= "purpose-id" DataType="String">
<AttributeValue> SecondOpinionOnTreatment</AttributeValue>
</Attribute>
<Attribute AttributeId= "compatibleWith" DataType= "bag">
</Attribute>
</Purpose>
</Target>
<Condition
<Function FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal"/>
 <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-is-in">
 -- Consent-Purpose is the purpose for which the data subject has consented or the purpose for which the data (Resource) is
legally stored on the grid database we have specified this purpose as an attribute of the resource in question--
<ResourceAttributeDesignator attributeId= Consent Purpose DataType = "string"/> BreastCancerDiagnosisAndTreatment
</ResourceAttributeDesignator>
</Apply>
<Apply>
<PurposeAttributeDesignator attributeId= "CompatibleWith"  DataType = "bag"/>
</Apply>
</Condition>
</Rule>
```

Figure 5: Privacy aware XACML rule.

7 SEMI-AUTOMATED MAPPING OF SWRL RULES TO XACML RULES

In order to further automate the mapping of SWRL rules to XACML rules, we rely on mapping templates where we can specify for each OWL property an equivalent XACML attribute ID. Furthermore, we need a detailed *one to one* mapping between the OWL axioms specifying conditions/constraints on the different elements of a rule target and the standard XACML functions that could be used as alternatives to these axioms once applied on XACML attribute-ids.

In most of the cases, an XACML equality function/predicate would be the relevant function to allow the translation of an OWL property constraint. The two operands of the equality are first, the *attribute_id* that should hold the name of the OWL property and second the attribute value that should be the same as the OWL property value. XACML distinguishes between several equality checking functions depending on the *data types* of the *operands*. The equality functions in XACML include *string-equal*, *Boolean-equal*, *Integer-equal* and other types. Deciding on which one we need to select is based on a mapping between the OWL *data type* of the property *value* and XACML *data types*. If the property value is determined by an *object property* then an XACML attribute matching function of type *string* should be used. If the property value is determined by a *data type property*, then the XACML attribute matching function should have the same type as the *data type property value*. The work presented in (Kolovski et al., 2006) and (Kolosvki et al., 2008) provides a detailed mapping of XACML data types to OWL data types. A reverse mapping is needed in our case, since we are interested in mapping OWL axioms to XACML conditions instead.

8 CONCLUSIONS AND FUTURE WORK

In this paper, we have modelled high level policies interpreted from European and national data protection law as privacy aware access control policies. The use of Semantic Web technologies such as OWL and SWRL allowed the integration of privacy requirements highlighted in text law such as requirements of consent and other safeguards of patient rights as policy constraints. Towards an easy enforcement in the security architecture of highly distributed infrastructures such as clouds, we have used mapping templates to transform the Semantic Web access control policies to a de facto and highly portable standard of access control notably XACML. The work is validated through the use of examples and scenarios of data processing and one of them is selected and presented in this paper i.e. *"Medical Images Exchange"*. This permitted to conclude that the use of ontologies and semantic technologies could provide relatively easy interpretation of legislation at an operational level. Few challenges were faced when conducting this work that we have overcome by mapping the SWRL privacy policies to XACML policies. An interesting future work in this area for us is to produce an extended XACML enforcement architecture that is able to adequate the added semantic layer for the SWRL to XACML mapping task. This will require both an implementation of the mapping formalism and testing it on the extended enforcement architecture.

REFERENCES

Brandic, I. and Dustdar, S. and Anstett, T. and Schumm, D. and Leymann, F. and Konrad, R., 2010. Compliant Cloud Computing (C3): Architecture and Language Support for User-Driven Compliance Management in Clouds, *IEEE 3rd International Conference on Cloud Computing.*

Bechhofer, S. et al., 2004. OWL Web Ontology Language Reference, [Online] *W3C* Available at: http:// www.w3.org/ TR/owl-ref/ [Accessed 2013].

Beyleveld D, Townend D., Rouillé-Mirza S., Wr ight J., 2004. Implementation of the Data Protection Directive in relation to medical research in Europe, *Ashgate Publishing Limited,* ISBN-10: 0754623696.

Boley, H. et al., 2010. Schema Specification of RuleML 1.0, [Online] Available at: http://ruleml.org/1.0/ [Accessed 2012].

Casassa Mont, M., Crosta, S., Kriegelstein, T. & Sommer, D., 2007. PRIME Architecture V2, *Deliverable D14.2.c.* [Online] Available at: https://www. primeproject.eu/prime_products/reports/arch/pub_del_ D14.2 .c_ec_WP14.2_v1_Final.pdf [Accessed 2013].

Casassa Mont, M., Shen, Y., Kounga, G. & Pearson, S., 2010. *EnCoRe Project Deliverable D2.1. Technical Architecture for the first realized Case Study.* [Online] (1.0) Available at: http://www.encoreproject.info [Accessed June 2013].

Damiani, E., di Vimercati, S. D. C., Fugazza, C. & Samarati, P., 2004. Extending Policy Languages to the Semantic Web, *in Proceedings of the 4th International Conference of Web Engineering*, Springer.

Demchenko, Y., Koeroo, O., de Laat, C. & Sagehaug, H., 2008. Extending XACML authorisation model to support policy obligations handling in distributed applications. *In Proceedings of the 6th International Workshop on Middleware for Grid Computing*, ACM.

EC.Directive 95/46/ECofthe European Parliament and of the Council.,1995 (cited 2010). *Available online from:* http://ec.europa.eu/justice/policies/privacy/law/index_en.htm#directive.

Gowadia, V., Scalavino, E., Lupu, E. & Aziz, B., 2008. The Consequence Project, *Deliverable D3.1: Models and framework for Meta-data generation and policy infrastructure.* [Online] Available at: http://www.consequenceproject.eu/ Deliverables_Y1/D3.1.pdf.

Gruber, T. R., 1995. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies*, 43(4-5),pp. 907--928.

Horrocks, I. et al., 2004. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. [Online] *W3C Available at:* http://www.w3.org/Submission /SWRL/ [Accessed 2013].

Italian Personal Data Protection Code, 2003. Legislative Decreeno. 196 of 30 June 2003. Online; 2003 (cited 2012). Available from: http://www.privacy.it/ privacycode-en.html.

Iversen A., Liddell K., Fear N., Hotopf M., Wessely S. Consent, 2006. Confidentiality and the Data Protection Act, *British Medical Journal (Clinical Research Ed)*, 332 (7534):165–169.

Jeremy W. Bryans, John S. Fitzgerald, 2007. Formal engineering of XACML access control policies in VDM++, *Proceedings of the formal engineering methods 9th international conference on Formal methods and software engineering*, November 14-15, Boca Raton, FL, USA.

Kolosvki, V. 2008. Logic-based Framework For Web Access Control Policies, *PhD Thesis, Digital Repository at the University of Maryland*, College Park, Md.

Kolovski, V. & Hendler, J., 2008. XACML policy analysis using description logics, [Online] *Available at:* http://www.mindswap.org/□kolovski/KolovskiXACMLAnalysis- JCSSubmission.pd [Accessed 2012].

Kolovski, V., 2006. Formalizing XACML Using Defeasible Description Logics. *Technical Report TR-233-11,* University of Maryland, College Park.

Masi, M., Pugliese, R., Tiezzi, F., 2012. Formalisation and Implementation of the XACML Access Control Mechanism, *In ESSoS. LNCS* 7159, 60-74, Springer.

Matteucci, I., Petrocchi, M. & Sbodio, M.L., 2010. CNL4DSA – a Controlled Natural Language for Data Sharing Agreements, *In Proceedings of the 2010 ACM Symposium on Applied Computing,* Sierre, Switzerland, ACM.

McCullagh, K., 2006. Study of data protection: harmonization or confusion? *In Proceeding of the 21st BILETA Conference: Globalisationand Harmonisation in Technology Law*, Malta.

Muppavarapu, V. & Chung, S.M., 2008. Semantic-Based Access Control for Grid Data Resources in Open Grid Services Architecture - Data Access and Integration (OGSA-DAI), *in 20th IEEE International Conference on Tools with Artificia lIntelligence (ICTAI 2008),* Dayton, Ohio, USA, 2008. IEEE Computer Society.

OASIS XACML, 2005. eXtensible Access Control Markup Language (XACML), Version 2.0 (2005), *Available online at* http://docs.oasisopen.org/xacml/2.0/XACML-2.0-OS NORMATIVE.zip.

OASIS XACML, 2013. eXtensible Access Control Markup Language (XACML), Version 3.0, http://docs.oasis-open.org /xacml/3.0/xacml-3.0-core-spec-os-en.pdf [Accessed 2013].

OCSI: The Open Cloud Standards Incubator, 2010. Architecture for Managing Clouds, *White Paper from the Open Cloud Standards Incubator* 1.0, DMTF DSP-IS0102, [Online] Available at: http://www.dmtf.org/ standards/published_ documents/DSP-IS0101_1.0.pdf.

Ontario, 2008. Freedom of Information and Protection of Privacy Act, [Online] Available at: http://www.elaws.gov.on.ca/html/statutes/english/elaws_statutes_90f31_e.htm [Accessed 2013].

Powers, C., Adler, S. & Wishart, B., 2004. EPAL Translation of the Freedom of Information and Protection of Privacy Act, *White Paper IBM Tivoli and Information and Privacy Commissioner,* Ontario.

Priebe et al., 2006. Mitigate Content-Related Risks With Enterprise Rights Management. *Trends. Forrester Research.*

Rahmouni H. B. , Solomonides T., Casassa Mont M, Shiu S., 2010. Privacy compliance and enforcement on European Healthgrids: an approach through ontology. *Philosophical Transactions of the RoyalSociety*, 368: pp 4057–4072.

Rahmouni H. B., Solomonides T., Mont M. Casassa, Shiu S, Rahmouni M. A., 2011, Modeldriven Privacy Compliance Decision Support for Medical Data Sharing in Europe. *Methods Inf Med.* 2011 Aug 15;50(4):326–36.

Rahmouni, H. B., Solomonides, T., Casassa Mont, M. & Shiu, S., 2011. Ontology Based Privacy Compliance for Health Data Disclosure in Europe. *PhD Thesis, University of the West of England*, Bristol, UK.

Sommer, D., Casassa Mont, M. & Pearson, S., 2008. PRIME Architecture V3. *Deliverable 14.2.d*. [Online] Available at: https://www.primeproject.eu/prime_products/reports/arch/pub_del_D14.2.d_ec_WP14.2_v3_Final.pdf [Accessed 2013]

Wang, K., Billington, D., Blee, J. & Antoniou, G., 2004. Combining Description Logic and Defeasible Logic for the Semantic Web. *In Rules and Rule Markup Languages for the Semantic Web: Third International Workshop*, RuleML. Hiroshima, Japan, Springer.