Modeling and Expressing Purpose Validation Policy for Privacy-aware Usage Control in Distributed Environment

Rath, Thavy Mony Annanda; Colin, Jean-Noël

Publication date:
2014

Document Version
Early version, also known as pre-print

Modeling and Expressing Purpose Validation Policy for Privacy-aware Usage Control in Distributed Environment

Annanda Thavymony Rath
PReCISE Research Center
Faculty of Computer Science, University of Namur, Belgium
rath.thavymony@unamur.be

Jean-Noël Colin
PReCISE Research Center
Faculty of Computer Science, University of Namur, Belgium
jean-noel.colin@unamur.be

ABSTRACT

Privacy-aware usage control is a control of the usage of private data with the aim to protect data owner privacy. In privacy-aware system, the purpose of data usage[1] is strictly controlled to ensure that data owner privacy is properly protected and data would never be used beyond what it is authorized for. To fulfill that level of protection, it requires the strong enforcement of usage policy, in particular, the enforcement of the purpose of data usage. However, there are many difficulties in purpose enforcement. One of which is to validate the purpose of an agent when it requests to perform an action, particularly in distributed environments where the processing of data is carried out on client side application and direct control of it is limited. Generally, validating "a particular purpose" may require different mechanisms and can happen at different points in time[2] during the lifecycle of data usage. Hence, there is a need to express "how purpose should be validated" by indicating which validation mechanisms should be used and when the validation should take place so that the remote system can act as instructed. In this paper, we discuss the design issue of purpose validation policy[3] expression based on our proposed validation structure: pre-, ongoing-, and post-validation. Furthermore, we discuss how the existing languages such as EPAL, XACML, and ODRL can directly be used or extended to support our proposed purpose validation policy model.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]

[1]Purpose of data usage is a purpose of using or accessing data.
[2]Time refers to a point in time during the usage of data where the purpose should be validated. For instance, before user is granted usage permission, during the usage of data, or after the usage of data.
[3]Purpose validation policy contains the rule and information provided to remote client application for "purpose" validation process.

IMCOM (ICUIMC)'14, January 9-11, 2014, Siem Reap, Cambodia.

General Terms

Theory

Keywords

Purpose enforcement, purpose validation policy expression, distributed system, security, privacy, usage policy, usage policy management.

1. INTRODUCTION

Purpose of usage is one of the core concepts in privacy which considers the data user's intent as a factor in making usage control decisions and the enforcement of it is required, to ensure that data is used as what it intends for [9][11][3][7][15].

In dictionary, "purpose" is defined as "the object toward which one strives or for which something exists; an aim or a goal". However, by observing how purpose is used in the natural language reveals that purposes often refer to an or a set of abstract actions [4] [5]. For example, accessing patient's health record for the purpose of treatment, research, insurance, etc., all of which are names of some abstract actions.

In general, the intentional actions are often referred to a purpose that expresses the aim to perform them. For example, physician accesses the patient's blood pressure record for heart surgery preparation, heart surgery operation, research, etc., hence, in everyday usages of purpose refers to a final aim concerning the goal behind performing an action or a set of actions and its ultimate consequences. Purpose may be different from areas to areas. For example, in e-health, "purpose" refers to patient treatment, research, insurance, etc.; while in other field like marketing we normally see the purpose often refers to actions such as: promoting, shipping, distributing information, etc.; all of which are names of some abstract actions.

In general, the enforcement of purpose [1][6][19] means to verify that those abstract actions exist and they are valid before data is released to requester; in some contexts, they also need to be valid during the usage of data [2][13][4]. There are two main parts for purpose enforcement.

1. First, "verification", a process to check that claimed purpose[5] exists for a given requested object and action.

[4]Abstract action is different from the concrete actions commonly used in processing of data like "read", "right", "view", etc. Abstract action is not really the action on resource, but the aims of using resource. For more detail, one can refer to [12].
[5]Claimed purpose is the purpose of using resource.

2. Second, "validation" refers to a process to prove that the claimed purpose is valid at the time of usage. For example, if physician claims "heart-surgery" as purpose of using patient health record, then, "validation" means to prove that

   - physician does have the right to use data for "heart-surgery";
   - claimed purpose can be achieved after usage permission is granted (in order words, we need to make sure that physician will surely use patient's record for the claimed purpose); and
   - physician can not use patient's record beyond the authorized purpose.

Validating purpose of action [14] is the main difficulty in purpose enforcement, particularly, in distributed environment where the source system has limited capability to directly control data at remote client application. How to instruct remote client to validate the purpose in accordance to the policy of source system is the main issue that needs to be addressed. To do so, source system needs to provide to remote client the Purpose Validation Policy (PV-Policy) that can be used for purpose validation process at client side control domain.

It is worth noting that PV-Policy is not the usage policy, which expresses how the resource should be used. PV-Policy expresses how the purpose validation should be performed. However, it is possible to embed PV-Policy into usage policy (detail of it can be found in the following sections). It is also important to note that, in this paper we address only the purpose validation policy expression in the context of distributed environment. We do not address the issue of purpose validation mechanism[6].

The rest of the paper is organized as following. Section 2 is about the motivating example. We present purpose validation policy structure and its model in Section 3 and the XML-based encoding of the model in Section 4. Section 5 is the discussion about the existing languages: XACML, EPAL, and ODRL. Section 6 talks about the purpose-based usage enforcement model and its prototype. Section 7 is the related work and contribution and Section 8 is the conclusion and future work.

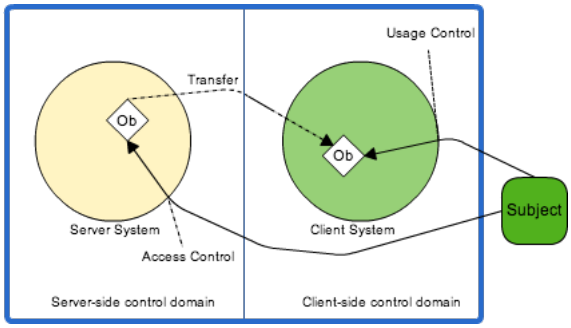

Figure 1: Illustration of access and usage control concept

[6]Validation mechanism is a technique used to validate the purpose of date usage.

2. MOTIVATING EXAMPLE

To make clear between the concept of access control (AC) and the concept of usage control (UC), we provide a brief definition of them. In general, two controlling steps are required to ensure that protected data goes to the right people and is used in the right directions. Those steps are access and usage control. These controlling steps happen in two different circumstances, one before another after data is granted access.

Access control[3], the main goal of it is to selectively determine who can access services, resources, and digital contents and what access is provided exactly. Access control prevents unauthorized access to the resources of system.

While access control concerns about who should be allowed to access, usage control concerns about what will and will not happen to data object once it is granted access.

As presented in Figure 1, subject (user) requests to access to object (Ob) from the server (server side control domain) and the object (Ob) is stored for later use in the client side system. Once subject is authenticated and verified, the data object (Ob) and its corresponding usage policy is transferred to client side system and the future usage of Ob is controlled by client side control domain. The issue is that once the data is at client system, server side system is no longer able to control it. The protection of data depends only on the client side system. Thus, a usage policy enforcement mechanism is required to ensure that client side system treats the data object in the same level of security as had been done by server side system, inline with the usage policy.

It is worth noting that although our ultimate goal is to enforce the privacy-aware usage policy, we address in this paper only a technique for providing "purpose validation policy" to the client side system for enforcing the usage policy. To be clear, what we address in this paper is purpose validation policy expression, one of the important issues for enforcing usage policy in distributed environment.

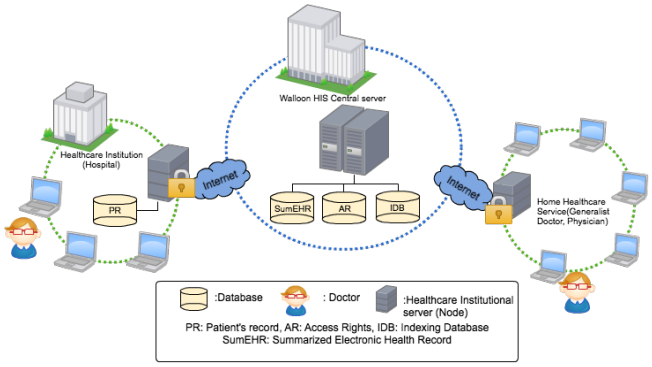

Figure 2: Simplified Schema of Walloon Healthcare Network

In order to illustrate the practical examples of our addressing issue, we use the example of distributed healthcare system. In this case, we take a real application scenario from Walloon Healthcare Network (WHN)[21]. It is worth noting that all the examples given in this paper are in context of distributed healthcare system.

As illustrated in Figure 2, the WHN is a network of health institutions such as hospitals, and clinics, but also of physi-

cians, that aims at supporting the exchange of patient's medical record between healthcare professionals, in a timely and secure way.

In this scenario, the access permission and usage policy are managed by WHN central server. Patient's electronic health records can be shared among different healthcare institutions in the network. The usage authorization on patient's record is strictly controlled and usage permission depends on the exact purpose of data usage (e.g., emergency treatment for heart-surgery, personal archive, or research purpose). Below are important requirements for controlling the usage of patient's health record.

1. Health record can reside on a user's device for limited period of time.
2. It can be shared among healthcare professionals.
3. The permission to reuse the record depends on the current state of purpose validation, not the validity of the purpose at the time of first access[7].

The third requirement means that although the purpose of usage is valid at the time of first access, the validity of it is no use for the later usage (re-use) at remote client. Revalidation of purpose is required. In other words, the usage authorization should be based only on the validity of purpose at time of usage, but not on the result of the first access. Thus, purpose validation is a continuous control process during the lifecycle of data usage at remote client rather then a once time control at the first access to data at source server. The purpose of usage should be checked before, during, and after the usage of data to ensure that the usage of data complies with the claimed purpose.

Thus, the question is how does the remote client perform the re-validation of purpose? The remote client should have some information such as what mechanism should be used and when the validation should take place given that different mechanism may be applied for different type of purpose and validation can take place at different phase in the lifecycle of data usage. This example shows clearly the necessity of having a way to provide such information to remote client so that it can perform correctly as instructed by source system.

3. PURPOSE VALIDATION POLICY STRUCTURE AND ITS MODEL

Observing how the data are processed in the real world reveals that there are three crucial states that need to be considered for purpose validation: before usage permission is granted, during the usage of data, and after using it. We term the three validation states as pre-, ongoing-, and post-validation respectively.

- Pre-validation refers to a validation state before granting usage permission to user. At this state, user's request is checked and purpose of usage is validated. If system finds that the claimed purpose is not valid, it rejects request immediately before usage session starts.
- Ongoing-validation refers to a validation state when user is using data. At this state, system continuously controls purpose of usage. It checks if the actions performed and the requesting actions are complied with the claimed purpose. During the usage session, system periodically triggers purpose re-validation process. This intends to check if the purpose of usage is still valid given the change of time or state.
- Post-validation refers to a validation state after the usage of data. It does not provide ongoing control of data usage, instead it provides a way to prove the correctness of the data usage by means of the log-information. Auditing mechanism is required to analyze the log-information and to reconstruct the execution process in order to find out if violation happened or not.

To support this validation structure, we propose a purpose validation policy model as follows. We define purpose as a tuple of PV (Purpose Validation) that consists of 4 elements.

PV= F(CP, VP, T, VM)

Where:

- "CP " is a claimed purpose of data usage.
- "VP" is a validation phase, it tells when the purpose should be checked, it can be "pre-, ongoing-, or post-validation".
- "T" is used for ongoing- and post-validation. When it is used in ongoing-validation, "T" refers to the time period for re-validating purpose. In case of post-validation, "T" refers to a time at which the purpose validation takes place after the data usage is ended.
- "VM" is Validation Mechanism, it describes the mechanism used to check the validity of claimed purpose.

In general, these four information (CP, VP, T, VM) are attached to data and they are sent to remote client. With the provided information, the remote client configures its system and validates purpose accordingly.

For example, PV=("heart-surgery", "pre-validation", "N/A", "Role-based purpose validation"), expresses that any request with the purpose of "heart-surgery" should be pre-validated by using the "role-based validation " mechanism.

Figure 3 presents a core purpose validation policy model, which consists of six entities presented below.

1. "Purpose validation" is the top-level entity containing the following attributes. "ValidationID" is the unique identification of the entity. "ValidationName" indicates name of purpose validation. "ValidationDescription" is a description of purpose validation.
2. " Purpose validation combining algorithm" is an entity providing the information on how the results from different purpose validation mechanisms should be examined. It contains the following attributes. "AlgoID" is the unique identification of the entity. "AlgoFunction" indicates how a purpose that requires multiple validation mechanisms should be validated. There are two types of function: "any of" and "all of " with the returns values: "Permit" and "deny-overrides" respectively. "any of " means, if the result of one mechanism is positive, the request is authorized while "all

[7]"Time of first access" refers to the time at which user requests access to source server before the data is transferred to remote client, Figure 1. At this state, the purpose validation may be performed at source server to prove the claimed purpose.

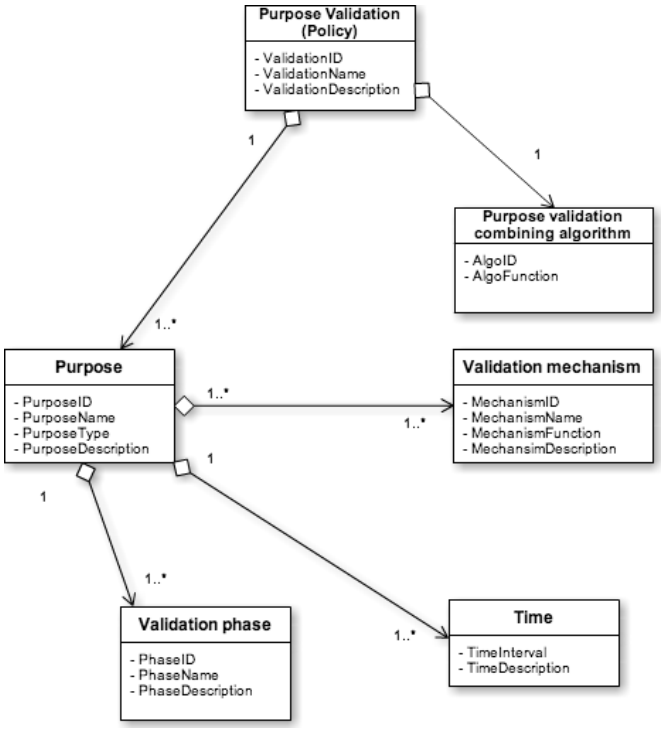

Figure 3: Core purpose validation policy model

of" means unless all the mechanisms provide positive result, the request is authorized.

3. "Purpose" is an important entity and contains the following attributes: "PurposeID" is the unique identification of the purpose entity. "PurposeName" indicates the name of purpose; for example, heart-surgery, research, or statistic . "PurposeType" indicates the type of purpose. It can be a normalType or HierarchicalType . "PurposeDescription" provides the detail information for purpose entity.

4. "Validation mechanism" is an entity, which is responsible for providing the information concerning the mechanism used for validating purpose, it contains the following attributes. "MechanismID" is the unique identification of the mechanism used to validate purpose. "MechanismName" indicates the name of purpose validation mechanism. For example, role-based [13], workflow-based [10], or data to action alignment validation mechanism. "MechanismFunction" indicates the function used to validate purpose. This function returns validation result with a "positive" or "negative" response. "MechanismDescription" provides the detail description of a particular mechanism.

5. "Validation phase" is an entity providing the information concerning when purpose validation should take place, it contains the following attributes. "PhaseID" is the unique identification for each phase of purpose validation. "PhaseName" indicates the name of validation phase. In this case, the attribute value can be "pre-validation", "ongoing-validation", or "post-validation". "PhaseDescription" provides the detail description about each validation phase.

6. "Time" is an entity providing the information about how often should the re-validation of purpose be taken place. This entity is designed to use in ongoing- and post-validation. "TimeInterval" indicates the series of time at which the re-validation of purpose should be performed. For example, the re-validation of purpose should take place every 10 minutes during the data usage session or after 10 days of data usage. "TimeDescription" provides the detail information on each time interval.

4. THE XML-BASED ENCODING

In this section we present the XML-based encoding of our proposed model. This specification starts with the description of the notations and terminology used in this document and follows by the explanation of XML encoding of each building block of the model. In order to improve the readability, we assume in this specification, the examples use the following XML internal entities declaration.

- PV is a name space that refers to Purpose Validation.
- $< pv : purposevalidation >$ refers to entity "purpose validation" in core model.
- $< pv : purpose >$ refers to entity "purpose" in core model.
- $< pv : validationmechanism >$ refers to entity "validation mechanism".
- $< pv : validationphase >$ refers to entity "enforcement phase".
- $< pv : time >$ refers to "Time" entity in core model.

Figure 4 shows a XML encoding of the model proposed in Figure 3. Figure 4 provides the information for validating a particular purpose called "heart-surgery", this purpose needs to be pre-validated by using two validation mechanisms: role-based and data to purpose alignment.

5. DISCUSSION XACML, EPAL, AND ODRL

In this section we discuss how the existing languages can be used to express our proposed purpose validation policy. We have studied three policy expression languages: XACML, EPAL, and ODRL. We starts this section with a brief introduction of each language and then go to its ability to handle our proposed purpose validation policy model. We will point out where our information can be embedded into their policy language so that it can be retrieved later for purpose validation at remote client application. We also point out if those languages need to be extended both its vocabularies and core model to support our proposed purpose validation policy model. It is important to note that we used XACML specification version 3.0, EPAL version 1.2, and ODRL version 2.0 as the main documents for our study.

5.1 XACML

XACML (eXtensible Access Control Language) is an OASIS [22] standard that describes both the policy language and an access control decision and response. The policy language is used to describe general access control requirements

```
0 <?xml version="1.0" encoding="UTF-8" ?>
1 <pv:purposevalidation
2 validationID= "purposevalidation001"
3 validationName="validate-heart-surgery-purpose"
4 validationDescription="It is requires for purpose to be validate before granting access to data"
5 algoID="purpose-validation-combining-algorithm"
6 algoFunction="deny-overrides">
7       <pv:purpose
8        purposeID="001"
9        purposeName="heart-surgery"
10       purposeType="hierarchicalType"
11       purposeDescription="heart-surgery is the high level purpose of heart-surgery-
                                 preparation " >
12           <pv:validationmechanism
13            mechanismID="M001"
14            mechanismName="role-based validation"
15            mechanismFunction="test:purposevalidation:role-based-validation-function"
16           mechanismDescription="this enforcement mechanism uses role to purpose
                                    alignment as  a medium to enforce the purpose" />
17             <pv:validationphase
18              phaseID="01"
19              phaseName="pre-validation"
20              phaseDescription="this purpose should be pre-validated, authorization to
                                    use data can happen only if purpose is valid"/>
21             <pv:time />
22    </pv:purpose>
23    <pv:purpose
24     purposeID="001"
25     purposeName="heart-surgery"
26     purposeType="hierarchicalType"
27     purposeDescription="purpose of heart-surgery-preparation " >
28           <pv:validationmechanism
29              mechanismID="M002"
30              mechanismName="data-to-purpose alignment"
31              mechanismFunction="purposevalidation:data-to-purpose-validation-function"
32              mechanismDescription="Purpose alignement" />
33            <pv:validationphase
34              phaseID="01"
35              phaseName="pre-validation"
36              phaseDescription="authorization to use data can happen  if purpose is valid" />
37            <pv:time  />
38      </pv:purpose>
39 </pv:purposevalidation>
```

Figure 4: A detail XML encoding for purpose validation policy

while the access control decision request/response language aims at providing the means to form a query to ask whether or not a given action should be allowed or denied.

The XACML policy language model consists of three mains components, which are policy set, policy, and rule. A policy set contains the target, rule-combining algorithm, obligation, and advice. Target specifies the set of requests to which it applies, it is generally declared by policy writer. Rule-combining algorithm specifies the procedure by which the results of evaluating the component policy are combined when evaluating the policy. Obligation specifies the obligation that user or system needs to perform before granting access to user.

For policy component, it consists of the same components and they have the same functionality as that of policy set. Those components are target, rule-combining algorithm, obligation, and advice.

For rule component, it consists of rule target, condition, obligation, and advice. "Condition" represents a Boolean expression that refines the applicability of the rule beyond the predicates implied by its target. For more detail, one can find in XACML specification[8].

Where "purpose of usage" can be expressed in the XACML policy? In general "purpose" can be expressed as a condition in rule. However, the expression of purpose as a condition in rule, is nothing more than to tell that the policy is applied on a particular purpose. It does not handle the validation. Since purpose validation can be considered as an obligation; hence, obligation can be used as a way to express purpose validation in XACML policy, but with a minor extension of the existing obligation structure. For example, allowing the obligation expression field $< ObligationExpressions >$ to have the purpose-validation-combining-algorithm. Other attributes required in our model can be expressed in $< AttributeAssignmentExpression >$ in XACML.

In Figure 5, we provide an example, how purpose validation can be expressed in XACML obligation. It is worth noting that this example uses the role-based validation mechanism for "heart-surgery" purpose with the "pre-validation" as the validation phase.

```
0   <ObligationExpressions>
1   <ObligationExpression ObligationId=
        "urn:oasis:names:tc:xacml:example:obligation:role-based purpose validation"
2        FulfillOn="Permit">
3   <AttributeAssignmentExpression AttributeId=
        "urn:oasis:names:tc:xacml:3.0:example:attribute:role-to-purpose-alignment">
4   <AttributeAssignmentExpression AttributeId=
        "urn:oasis:names:tc:xacml:3.0:example:attribute:validation_phase">
5   <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
                                    pre-validation</AttributeValue>
6   </AttributeAssignmentExpression>
7   <AttributeAssignmentExpression AttributeId=
        "urn:oasis:names:tc:xacml:3.0:example:attribute:user-role">
8   <AttributeValue DataType=
        "http://www.w3.org/2001/XMLSchema#string">physician</AttributeValue>
9   </AttributeAssignmentExpression>
10  <AttributeAssignmentExpression AttributeId=
        "urn:oasis:names:tc:xacml:3.0:example:attribute:purpose-of-usage">
11  <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
                                    heart-surgery</AttributeValue>
12  </AttributeAssignmentExpression>
13  </ObligationExpression>
14  </ObligationExpressions>
```

Figure 5: An example of purpose validation expressed in XACML obligation

5.2 EPAL

EPAL (Enterprise Privacy Authorization Language) is a formal language for writing enterprise privacy policies to govern data handling practices in IT systems according to fine-grained positive and negative authorization rights. It concentrates on the core privacy authorization while abstracting data models and user-authentication from all deployment details such as data model or user-authentication.

EPAL policy defines lists of hierarchies of data-categories, user-categories, and purposes, and sets of actions, obligations, and conditions. "User-categories" are the entities that use collected data. "Data-categories" define different categories of collected data that are handled differently from a privacy perspective. "Purposes" model the intended service for which data is used. "Actions" model how the data is used. "Obligations" define actions that must be taken by the environment of EPAL. "Conditions" are Boolean expressions that evaluate the context. These elements are then used to formulate privacy authorization rules that allow or deny actions on data-categories by user-categories for certain purposes under certain conditions while mandating certain obligations.

In EPAL, the "purpose" is part of EPAL authorization query. Without knowing the purpose of an access, authorization cannot be decided. As a consequence, any system using EPAL must be able to determine a purpose before asking the EPAL engine to evaluate a given policy. This means that EPAL policy does not directly handle the purpose vali-

[8]http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-en.html

dation expression. Determining the purpose depends on the application. Privacy-aware applications may even state their current purpose or an activity identifier that can be used to easily derive it. For more detail information, one can go to EPAL specification[9].

Like XACML, in EPAL, purpose validation policy can be expressed in obligation. The required information in our proposed model can be expressed by using $< parameter >$ in obligation where "purpose-validation-combining-algorithm" can be defined as a rule in obligation. Figure 6 provides an example of purpose validation policy expression in EPAL obligation for purpose as "heart-surgery" with role-based as purpose validation mechanism. "Pre-validation" is used in this example.

```
0 <obligation refid="heart-surgery validation">
1   <originating-rule refid="role-based purpose validation"/>
2    <parameter refid="validation-phase"  simpleType=
                          "http://www.w3.org/2001/XMLSchema#string">
3    <value>pre-validation</value>
4    </parameter>
5    <parameter refid="user-role" simpleType=
                          "http://www.w3.org/2001/XMLSchema#string">
6      <value>physician</value>
7    </parameter>
8    <parameter refid="purpose" simpleType=
                          "http://www.w3.org/2001/XMLSchema#string">
9      <value>heart-surgery</value>
10  </parameter>
11 </obligation>
```

Figure 6: An example of purpose validation expressed in EPAL obligation

5.3 ODRL

ODRL (Open Digital Rights Language) is a standardized language for Digital Rights Management (DRM) community to express the rights information over content. The ODRL aims at providing the flexible and interoperable mechanism to support the use of the digital resources in publishing, distributing and consuming of the electronic publication, digital images, learning object, computer software, and other digital forms. ODRL 2.0 consists of central entity (policy) interconnecting with other entities such as permission, prohibition, duty, party, asset, action, and constraint.

In ODRL 2.0, the purpose of access is expressed in constraint. ODRL is not designed to particularly deal with the privacy data like EPAL. Comparing to EPAL and XACML, ODRL has less capability to express purpose validation expression although the "duty" entity, which is equivalent to "obligation" in EPAL or XACML may be used to express purpose validation policy. In ODRL, "duty" consists of four entities. Those are asset, action, constraint, and party. "action" refers to obligation functions that user or system needs to perform. "asset" refers to resource used to fulfill duty. "party" refers to the parties involved in fulfilling the duty. "Constraint" is used to provide a fine-grain expression for duty.

Given the current structure of "duty" in ODRL 2.0, it is hard to use it to embed our proposed purpose validation policy. A major re-structuring of the "duty" structure as well as the vocabulary used in "duty" is required in order to support our proposed purpose validation policy model. For more detail information about ODRL, one can refer to ODRL specification[10].

5.4 Discussion

Using the existing languages to express the proposed purpose validation policy is a way to support the existing standard model. However, it is not the only way we can use to express the purpose validation policy to remote client. Since those languages consider purpose validation as out of their scope, we can have our separate file expressing the purpose validation policy as shown in Figure 4. Languages such as XACML, EPAL, or ODRL, clearly mention that their mission is only to express how data should be processed and validation of purpose is the role of the application developer. For example, although EPAL is designed for privacy policy where purpose is an important factor in an authorization, it does not handle the validation part, any system using EPAL must be able to determine a purpose of using data before asking the EPAL engine to evaluate a given policy.

With this reason, other alternative for expressing the purpose validation policy is to use the separate file for it without touching the usage policy expression file of the existing language. This means that the remote client no longer has "data + usage policy" in the package from source server, instead, it has "data + usage policy + purpose validation policy". With this new concept, remote client application developer must create a separate "purpose validation" module, which is responsible for validating purpose of usage and it is able to retrieve the validation information expressed in purpose validation policy. This module can be integrated into the existing engine of other languages such as "enterprise-java-xacml". The module will act as the frontline, validates purpose and decides before passing the usage policy for further evaluation by existing standard engine.

6. PURPOSE-BASED USAGE ENFORCEMENT MODEL AND PROTOTYPE

In this section, we present a purpose-based usage enforcement model and prototype implementation. This proposed model aims at applying our proposed purpose validation policy model. We build the purpose validation module which is integrated into the existing engine of other language, in this case, we use enterprise-java-xacml. The enforcement model focuses on the system architecture and functional modules to illustrate how the purpose validation can be achieved with a given purpose validation policy. It is worth noting that the proposed purpose enforcement system architecture and the prototype are primarily designed for pre-validation of purpose. For ongoing- and post-validation, we address them in our future work.

6.1 Purpose enforcement model

As illustrated in Figure 7, the model consists of the following components.

1. EP handles the requests from subject and forwards those requests to Purpose Validation Point and then to decision point for further policy evaluation.

[9]http://www.w3.org/Submission/2003/SUBM-EPAL-20031110/

[10]http://www.w3.org/community/odrl/two/model/

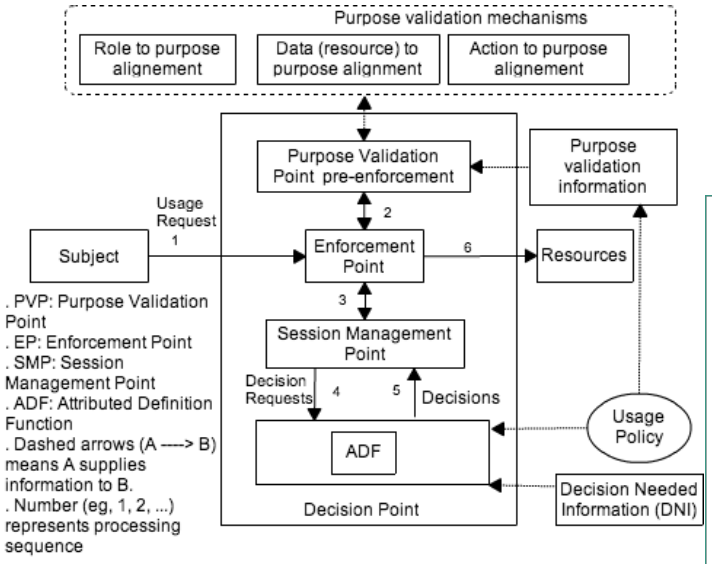

Figure 7: Purpose-based usage control enforcement model

2. SMP is the dynamic part of the whole engine and captures the continuity behavior of the usage control system. Furthermore, it manages the functions of other elements of the architecture and ensures the transitions from one to another state.

3. DP is responsible for making the required decision during a usage control session based on usage policy.

   - Attribute Decision Function (ADF) handles the attribute-based usage decision during a usage session. Attributes can be either subject, object, or environment attributes (e.g., the subject's identification). The information required by ADF is retrieved from DNI.

4. PVP makes the decision whether a purpose is valid or not. Whenever there is a request, PVP checks the request based on the claimed purpose. To validate the usage purpose, PVP retrieves the "purpose validation policy" that tells which purpose validation mechanism should be used and when the validation should take place. In our example we use three purpose validation mechanisms as presented in Figure 7: "Role to purpose alignment", "Action to purpose alignment", and "Data to purpose alignment".

   - "Role to purpose alignment" provides the information concerning the alignment between the requester's role and the purpose of access. For example, a requester in role of "cardiologist" may be aligned to the "heart surgery purpose". This information can be used for the pre-enforcement of purpose.
   - "Action to purpose alignment" provides the information concerning the alignment between the actions on object and the purpose. For example, action "transfer" is aligned to the "emergency purpose".
   - "Data to purpose alignment" provides the information concerning the alignment between type of object (resource or data) and purpose. For example, data concerning surgery may be aligned to the request for "surgery purpose".

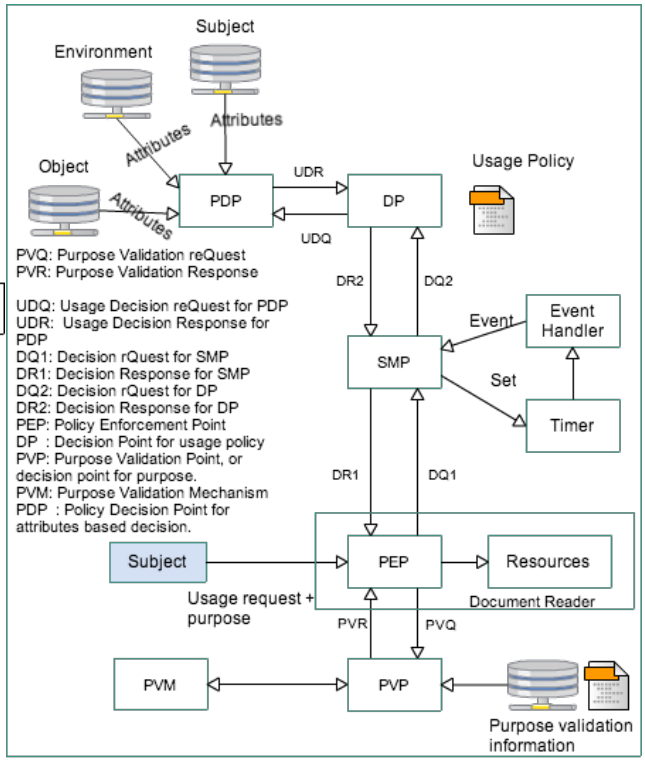

Figure 8: Implementation architecture

6.2 Model implementation (prototype)

To show the functionality of our proposed purpose enforcement model, we designed a concrete usage control enforcement engine as presented in Figure 8. We then developed and validated prototype in Java. Furthermore, in order to utilize and facilitate the existing standards and frameworks, a modified XACML "enterprise-java-xacml"[11] is used in this prototype as a core policy evaluation engine combined with our designed purpose validation engine.

It is worth noting that the model implementation aims only at showing the usefulness of our proposed purpose validation policy expression. Thus, the focus will be the design of Purpose Validation Point (PVP) and its associated components.

We take a usage scenario in healthcare domain. In this scenario, the doctor requests patient's health record from the Healthcare Information System (HIS). After authenticating and authorizing the doctor based on his role and purpose of usage; HIS releases the record, usage policy, and purpose validation policy in one package. The package can reside on the doctor device for a specific period of time during which doctor can re-use it. The enforcement component, which is integrated into the document reader, checks the integrity of

[11] http://code.google.com/p/enterprise-java-xacml/

the package and extracts the usage control policy, purpose validation policy, and patient's record. Figure 8 shows the architecture of our purpose-based usage control enforcement engine. The engine consists of the following components.

- PEP acts as single entry point to protected resources and performs usage control. It receives usage requests from subject, and first makes a purpose validation request (PVQ) and consequently receives the response (PVR) from PVP. After receiving the positive response from PVP, it makes a usage decision request (DQ) to PDP and gets the decision response (DR).

- PVP acts as a validation point for purpose. First, it checks which validation phase should be applied to the claimed purpose, either pre-, ongoing-, or post-validation. Then, purpose of usage is validated before further validation of usage policy by PDP. If PVP provides negative response, the process is ended here and no further evaluation of usage policy. In case of positive response from PVP, the further decision request is sent to PDP for further usage policy evaluation.

- PVM provides all the necessary validation mechanism to PVP in according to the information provided in purpose validation policy.

- PDP refers to ADF function in our enforcement model and is represented as an XACML PDP. It is the component that evaluates attribute related constraints (authorizations and conditions) and renders decisions to DP.

- Event handler handles the events that trigger transitions from one state to another. It listens to the events and sends the trigger actions to SMP when state change is about to occur.

- Timer can be set by the SMP through the event handler. Timer can be used for supporting re-validation process.

7. RELATED WORK AND CONTRIBUTION

Purpose is raised and argued in many literatures as an important entity used to control access to sensitive private data [20][16][14][12][7][15]. Byun et al [7][16] proposed a purpose-based access control of complex data for privacy protection, a model that relies on the well-known RBAC [8] access control model as well as the notion of conditional role which is based on the notion of role attribute and system attribute. In their paper, they provided also a general purpose tree applied in complex data management system and the solution to address the problem of how to determine the purpose for which certain data are accessed by a given user.

Jafari et al [12] defined a semantic model for purpose, based on which purpose-based privacy policies can be meaningfully expressed and enforced in a business system. The model is based on the intuition that the purpose of an action is determined by its situation among other inter-related actions. Actions and their relationships can be modeled in the form of an action graph. A modal logic and model checking algorithm are developed for formal expression of purpose-based policies and verifying whether a particular system complies with them.

Concerning enforcement, Katt et al [14] proposed the extension of $UCON_{ABC}$[17][23][18] with continuous control usage sessions for expressing the ongoing-check obligation. They also proposed the general, continuity-enhanced policy enforcement engine for usage control applied particularly to obligation. After the thorough study on the work of Katt et al, we found that the model can be extended and used to enforce the ongoing-validation of purpose, one of our proposed validation phase.

Jafari et al [1] proposed an approach to enforcing purpose in access control systems that uses workflows. They proposed to encode purposes as properties of workflows used by organizations. However, the proposed model does not work with "purpose" that does not have a natural interpretation in terms of workflows, particularly, more abstract purposes.

Other proposed mechanisms for purpose management and enforcement are self-declaration in which the agent explicitly announces the purpose of data access [7] and role-based enforcement [13] in which the purpose is identified based on the agent's role in the system. The first method obviously cannot stop a malicious agent from claiming false purposes. This is because anyone can claim any purpose of access, without the proper system to validate claimed purpose, this method cannot be used in sensitive private data processing environment like distributed healthcare [4]. The second method has been criticized to be inefficient in capturing purpose of an action since roles and purposes are not always aligned and members of the same organizational role may practice different purposes in their actions.

While many researches focus on validation mechanisms, there is a lack of discussion on how the purpose validation policy should be expressed in the context of distributed environment. This serves as our main addressing issue.

There are four contributions in this paper. First, we proposed the purpose validation structure. This structure provides to the policy writer a flexible way for determining the validation phase and mechanism used for particular purpose. Second, build upon these validation structure, we proposed a purpose validation policy model for privacy-aware usage policy. Third, we proposed the purpose-based usage control system architecture that takes into account the enforcement of purpose for distributed environment. Fourth, we implemented a prototype in Java as a step into validation of our proposed model.

8. CONCLUSION AND FUTURE WORK

In this paper, we outlined the issue of Purpose Validation Policy (PV-Policy) expression in distributed environment. We proposed the PV-Policy model. We also discussed on how PV-Policy can be expressed in the existing policy languages such as EPAL, XACML, and ODRL. According to our study, we found that obligation field (or entity) may be used to express those information; however, the extension both for model and their vocabulary are required since those languages are not built to address purpose enforcement.

To show the functionality of our proposed model, we built a prototype of the proposed purpose-based usage enforcement model and the purpose validation policy model. The purpose validation policy model is designed based on our proposed validation structure: pre-, ongoing-, and post-validation. However, in our prototype implementation, we addressed only the pre-validation while ongoing- and post-validation are left for the future work.

9. REFERENCES

[1] Mohammad Jafari, Reihaneh Safavi-Naini, and Nicholas Paul Sheppard . Enforcing purpose of use via workflows. In *Proceedings of the 8th ACM workshop on Privacy in the electronic society*, WPES '09, pages 113–116, New York, NY, USA, 2009. ACM.

[2] Alexander Pretschner , Hilty Manuel, and Basin David. Distributed usage control. *Commun. ACM*, 49:39–44, September 2006.

[3] Alexander Pretschner, Manuel Hilty, Florian Sch, Christian Schaefer, and Thomas Walter. Usage control enforcement: Present and future. *IEEE Security and Privacy*, 6:44–53, 2008.

[4] Annanda Th. RATH and Jean-Noël Colin. Patient privacy preservation: P-RBAC vs OrBAC in patient controlled records type of centralized healthcare information system. case study of walloon healthcare network, belgium. *The Fourth International Conference on eHealth, Telemedicine, and Social Medicine eTELEMED 2012*, 4:111–118, 2012.

[5] Annanda Th. RATH and Jean-Noël Colin. A purpose model and policy enforcement engine for usage control in distributed healthcare information system. 2013. HEALTHINF: 7th International Conference on Health Informatics, Barcelona, Spain, 2013.

[6] Annanda Th. RATH and Jean-Noël Colin. Towards purpose enforcement for privacy policy in distributed healthcare. pages 881– 886, 2013. CeHPSA - 2013 : 3nd IEEE International Workshop on Consumer eHealth Platforms, Services and Applications, Las Vegas, USA.

[7] Byun Ji-Won, Bertino Elisa, and Li Ninghui. Purpose based access control of complex data for privacy protection. In *Proceedings of the tenth ACM symposium on Access control models and technologies*, SACMAT '05, pages 102–110, New York, NY, USA, 2005. ACM.

[8] D.F.Ferraiolo, R.Sandhu, S.Gavrila, D.R.Kuhn, and R.Chandramouli. Proposed NIST Standard for Role-Based Access Control. In *ACM Transactions on Information and System Security*, pages 4(3):222–274, August 2001.

[9] EUdirective. *Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data.* https://www.cdt.org/privacy/eudirective/EU%Directive%.htmlHD%NM%1. Latest access: March 2013., 1995.

[10] Giovanni Russello, Changyu Dong, and Naranker Dulay. A workflow-based access control framework for e-health applications. *Advanced Information Networking and Applications Workshops, International Conference on*, 0:111–120, 2008.

[11] Healthcare Information and Management Systems Society. *The direction concerning the healthcare data in european countries.* http://www.himss.org/content/files/CPRIToolkit/version6/v7/D33_European_Union_Privacy_Directive.pdf, 2007.

[12] Jafari Mohammad, Fong Philip , Safavi-Naini Reihaneh, Barker Ken, and Sheppard Nicholas Paul. Towards defining semantic foundations for purpose-based privacy policies. In *Proceedings of the first ACM conference on Data and application security and privacy*, CODASPY '11, pages 213–224, San Antonio, TX, USA, 2011. ACM.

[13] Jawad Mohamed, Alvarado Patricia Serrano, and Valduriez Patrick. Design of priserv, a privacy service for dhts. In *Proceedings of the 2008 international workshop on Privacy and anonymity in information society*, PAIS '08, pages 21–25, New York, NY, USA, 2008. ACM.

[14] Katt Basel, Zhang Xinwen, Breu Ruth, Hafner Michael, and Seifert Jean-Pierre. A general obligation model and continuity: enhanced policy enforcement engine for usage control. In *Proceedings of the 13th ACM symposium on Access control models and technologies*, SACMAT '08, pages 123–132, New York, NY, USA, 2008. ACM.

[15] Lorenzo D. Martin, Qun Ni, Dan Lin, and Elisa Bertin. Multi-domain and privacy-aware role based access control in e-Health. *IEEE, Second International Conference on Pervasive Computing Technologies for Healthcare*, (10090047):131 – 134, Jan. 30 2008-Feb 2008.

[16] Ni.Qun, Bertino Elisa, Lobo Jorge, Brodie Carolyn, Clare-Marie Karat, and Trombeta Alberto. Privacy-aware Role-Based Access Control. *ACM Transaction Information and System Security*, 13:24:1–24:31, July 2010.

[17] Park Jaehong and Sandhu Ravi. Towards usage control models: beyond traditional access control. In *Proceedings of the seventh ACM symposium on Access control models and technologies*, SACMAT '02, pages 57–64, New York, NY, USA, 2002. ACM.

[18] Park Jaehong and Sandhu Ravi. The uconabc usage control model. *ACM Trans. Inf. Syst. Secur.*, 7:128–174, February 2004.

[19] A. T. Rath and J.-N. Colin. Towards purpose enforcement model for privacy-aware usage control policy in distributed healthcare. *Int. J. Secur. Netw.*, 8(2):94–105, Aug. 2013.

[20] M. C. Tschantz, A. Datta, and J. M. Wing. Formalizing and enforcing purpose restrictions in privacy policies. In *IEEE Symposium on Security and Privacy*, pages 176–190. IEEE Computer Society, 2012.

[21] WHN, 2009. Espace dévelopeur de RSW (Walloon Healthcare Network): https://www.reseausantewallon.be, latest access: July 2013.

[22] XACML:. *eXtensible Access Control Makeup Language,version 3.0.* http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-en.html, latest access: January-2013.

[23] Zhang Xinwen, Parisi-Presicce Francesco, Sandhu Ravi, and Park Jaehong. Formal model and policy specification of usage control. *ACM Trans. Inf. Syst. Secur.*, 8:351–387, November 2005.