Law-Aware Access Control: About Modeling Context and Transforming Legislation

Michael Stieghahn and Thomas Engel

University of Luxembourg, 6, rue R. Coudenhove-Kalergi, L-1359 Luxembourg
{michael.stieghahn,thomas.engel}@uni.lu

Abstract. Cross-border access to a variety of data defines the daily business of many global companies, including financial institutions. These companies are obliged by law and need to fulfill security objectives specified by legislation. Therefore, they control access to prevent unauthorized users from using data. Security objectives, for example confidentiality or secrecy, are often defined in the widespread eXtensible Access Control Markup Language that promotes interoperability between different systems.

In this paper, we show the necessity of incorporating the requirements of sets of legislation into access control. To this end, we describe our legislation model, various types of contextual information, and their interrelationship. We introduce a new policy-combining algorithm that respects the different precedence of laws of different controlling authorities. Finally, we demonstrate how laws may be transformed into policies using the eXtensible Access Control Markup Language.

1 Introduction

Although research on access control has been a topic of interest for years, the new field of *Legal Engineering* [4], in combination with access control, is of increasing importance. In times of an ongoing global financial crisis, an increasing demand for regulation of financial markets exists. Currently used remote desktop solutions, such as Citrix XenApp, VNC, or NX Nomachine, provide the convenience of a known desktop environment for their users. Such solutions are necessary because traveling employees of global working companies need access to data stored on the servers of their company. However, such remote desktop solutions do not dynamically restrict access to information that is necessary to fulfill a certain task but give full access to data. Similarly, today's access control systems (e.g. access control lists (ACL) and role-based access control (RBAC)) lack the possibility of including legal constraints in their access decisions. Nevertheless, deciding whether an access to specific data under a given context is legal is an indispensable factor for many companies.

We illustrate the necessity for a law-aware access control that incorporates legislation in an international banking application scenario using the following example, which is derived from results of interviews with bank consultants:

Example 1. A consultant travels by plane from country S to a customer located in a country D_1. The legislation of S comprises laws regarding bank secrecy and data protection. The destination country D_1 has a law that concedes the right to privacy; however, it has a restriction of this privacy that allows the border security to check mobile

K. Nakakoji, Y. Murakami, and E. McCready (Eds.): JSAI-isAI, LNAI 6284, pp. 73–86, 2010.

devices regarding their content. Therefore, airport security potentially checks the mobile device[1] and so, to avoid disclosure of confidential information, such data cannot be stored on the device. However, when meeting the customer, the consultant needs to access the data of the customer. Since bank secrecy and privacy can prohibit the use of a remote desktop solution in country D_1, the necessary data has to be transferred in advance to the device after the consultant has left the airport. An active connection possibly reveals a link between a customer and a bank. This breaks bank secrecy. Thus, a remote desktop solution might be the right choice, if the consultant and the customer could instead meet in country D_2, where the legal restrictions are not as strict as in D_1.

The legislation of a country litigates for everyone located within the country. However, accessing data such as confidential customer-related data or strategic information that is hosted in another country introduces the problem of being subject to at least two sets of legislation. The legislation of different countries may vary in respect of bank secrecy, data security, data privacy, cryptography, etc. Therefore, the access control system has to ensure a law-compliant access.

Various approaches, which extend RBAC by a variety of notions of context to overcome its limitations regarding dynamically changing situations, have been widely studied.

Bertino et al. introduce in [1] temporal authorization in a discretionary access control (DAC) system to combine authorization together with start and an expiration time. This approach supports temporal constraints, as we know from the time defined by the task when the data access is required and from the legislation at which time the access is legal. However, the time alone does not reveal whether or not an access to data is legal.

Strembeck and *Neumann* present in [11] an approach to enforce contextual constraints in a dynamic RBAC that checks the current values of contextual attributes for predefined conditions. In their approach, permissions can be associated with context constraints.

Damiani et al. define in [2] the spatially-aware access-control model GEO-RBAC. It enhances RBAC with spatial- and location-based information to model objects, user positions, and roles that are activated based on the position of the user. In add a physical position, users are also assigned a logical and device-independent position. However, binding the activation to roles based on the location information of the user is not sufficient when cross-border data access to confidential data is necessary. Thus, the location information can be used to serve two purposes: first, the location information for the start point and the end point of a connection is used to select the observable set of legislation, and second, it can localize a data access to a specific location to fulfill law-compliance.

Ungureanu and *Minsky* [13] and *Serban et al.* [8] describe a mechanism called Law-Governed Interaction (LGI) that regulates the activities of the participants in an e-commerce transaction. LGI allows participants, who are combined to a so-called open group of distributed heterogeneous agents, to interact with each other with confidence that this interaction is policy-compliant. The policies are called the law of the open group. In contrast to our solution, the term "location" means that laws are defined

[1] As happened recently:
`http://www.theregister.co.uk/2009/08/11/ripa_iii_figures/`,
`http://www.theregister.co.uk/2009/11/24/ripa_jfl/`

globally but enforced locally. Therefore, location is restricted to a group, a membership in a group, and contracts between participants, such that laws exist that are only valid for certain groups and not globally for all participants. Our solution uses location in the sense of a real location (a specific country or city as well as the proximity of a specific user). We also do not need a means for binding laws to certain users but bind instead to a location itself, because laws are enforced on the basis of the location.

1.1 Approach and Contribution

This paper reports our ongoing research to develop a law-aware access control system. We extend the approach introduced in [9], where we used a logic-based implementation, and in [10], where we used the eXtensible Access Control Markup Language (XACML). In this paper, we demonstrate how the widely used eXtensible Access Control Markup Language can be used to enhance an access control system. Today, XACML has become a *de facto* standard for access control policies. It is widely used to define policies that regulate access to data by providing a standard for access permissions as well as for access requests and their responses. Our contribution is to use the eXtensible Access Control Markup Language to incorporate legislation into access decisions by enriching policies with legal constraints. Those constraints are based on different types of context and their interrelations. By including legislation directly into access decisions, lawfulness can be ensured. To prevent overregulation, our approach guarantees that the access restrictions are only as strict as it is obliged by the legislation of the source and destination country.

1.2 Difference between a Health Care scenario and a Banking Scenario

Bank secrecy obliges financial institutions to secure data, which was given, for example, to provide a service, with appropriate security measures. For a data access from a location different from the head office, a financial institution has to guarantee that a data storage and a data processing is at least as secure as for a local access. Granting access from the outside may open unknown security holes and, therefore, this may compromise the security of this data. Applying the same access rights as for a local access is not sufficient to ensure the security of data, because usually a user is able to access more data than needed for a particular task. Financial institutions, however, have to ensure the security of accessed data and the best practice for the security is to limit access to necessary data. This does not imply that the risk of data leakage itself is decreased, but it minimizes the amount of data that may be lost.

Common examples of access control systems that deal with sensitive data are emergency scenarios, healthcare scenarios, and banking scenarios. A bank scenario differs from the two other scenarios in a fundamental way. The first priority in a bank scenario is to secure data against unauthorized access, data disclosure, and data loss. If this can be fulfilled, the second priority, the service to the customer, may be applied. In contrast, a healthcare scenario and, to an even greater degree, an emergency scenario rates safety over security, because saving life is more important than data security. If no service can be provided due to security reasons, it is inconvenient in the first case, but unacceptable in the second case. Therefore, an access control system for financial environments rather denies than grants access to data in a case of uncertainty.

1.3 Organization of the Paper

The remainder of this paper is organized as follows: In Section 2, we briefly describe the eXtensible Access Control Markup Language and how an XACML system decides about access. In Section 3, we describe our legislation model. To this end, we specify the different types of context information that are needed to incorporate legislation into access decisions. We introduce a new policy-combining algorithm that respects the precedence of different sets of legislation, which may overrule each other. Then, we describe how laws may be manually transformed into XACML policies. Finally, we describe briefly how an access decision is made by a system that follows our approach. Section 4 concludes the paper and outlines future work.

2 XACML

The eXtensible Access Control Markup Language is a declarative access control policy language designed to support authorization systems. XACML is implemented in XML to provides a processing model, describing how to interpret the policies and, as a second part, a request / response context language.

XACML [5] policies are structured as a tree of sub-policies (Fig. 1). Each tree has a defined target and a set of leaves containing a set of rules. A target defines certain conditions to determine whether this policy is applicable to a request. It is specified by four properties: a subject, a resource, an action, and an environment. *Subject* defines a user or process that requests access to a *resource*, which might be a file, a system, or a service. An operation on a resource is defined as *action*. *Environment* defines a set of attributes, which are necessary to decide about access, but which are not related to a specific subject, an action, or an environment. Attributes are features of a subject, a resource, an action or an environment. Rules define how to process a target and consist of Boolean expressions, which are interpreted and executed by the Policy Decision Point (PDP). Rules consist of a target, an effect, and conditions. The latter describe the state of the attributes of the target to satisfy the rule, whereas effect specifies how to proceed (e.g. *permit* or *deny*) if the conditions are satisfied. The response to the request is structured as follows: decision, status, and obligation. There are four possible decisions: *permit*, *deny*, *not applicable*, or *indeterminate*. Not applicable is returned if no rules or applicable policies can be found. Indeterminate indicates that an error occurred during

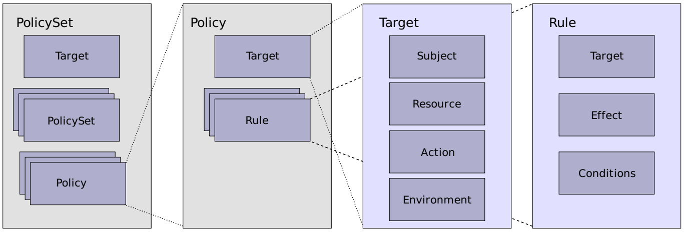


Fig. 1. XACML policy structure

the access decision. Obligations can be attached to the response and direct the Policy Enforcement Point (PEP), for example, to process an access in a designated way. However, XACML does not specify the communication protocol between PEP and PDP.

The XACML specification [5] defines additionally four combining algorithms to specify how policies of a policy set are combined during an access decision process.

2.1 Access Decisions

When connecting to a server, users need to authenticate themselves. Then they can use, for example, a file browser to browse a directory or to open a file. The client sends all actions as XACML requests to the PEP. The PEP resubmits the user's request to the Policy Decision Point. After receiving the request, the PDP starts to evaluate the top-level policy or the policy set. First, the target is checked whether the request matches the target specified in the policy. If a policy is evaluated to false, this policy is not applicable and a further evaluation of this policy is not necessary. Subsequently, the resources and actions specified in the target are evaluated as well. If the PDP evaluates the target to true the policies and rules in the next level below are evaluated. When the evaluation gets to a leaf of the XACML policy tree, the rule's conditions are executed. Every rule has an effect (permit or deny), which is sent back as decision if the condition is evaluated as true. Otherwise a non applicable is returned. The evaluation is performed with respect to the combining algorithm, which is specified for a policy or policy set and defines how the policies and rules need to be processed. During the evaluation, the PDP queries the attributes of the XACML request from the PIP, which collects subject, resource, environment, etc. On completion, the PIP sends the response to the PDP, which can then decide about the access. Finally, the PDP sends its access decision back to the PEP by using the XACML response language. The PEP executes the obligations bound to the policy and sends the final decision (permit / deny) to the user. If the system grants the access, the user is able, for example, to browse files on the file system or using a remote desktop solution.

3 Law-Awareness and Access Control

Remote access within a country, but especially cross-border access, implies that at least one set of legislation needs to be observed. In particular, financial institutions need to ensure both law-compliant access, which includes securing data against attacks, and serviceability. In general, laws define, among other things, conditions to satisfy and describe the handling of data and the access to data. When requesting access to data, a variety of context information can be used to support access decisions. In our approach, we mainly follow the definition of *Dey et al.* [3] for categories of context. In addition to their context types *identity*, *activity* (we use the term *task* as equivalent), *time*, and *location*, we extend the context by *legal constraints* and a *second identity*. Consequently, we describe our set of context information as: *Who* does *What* for *Whom*, *When*, *Where* and subject to *Which* legal constraints.

To improve readability, we elide the prefix *urn:oasis:names:tc:xacml:1.0:* in the urn-definitions of all policies in this document. For the same reason, we also elide *http://www.w3.org/2001/XMLSchema#* from the definition of the data type.

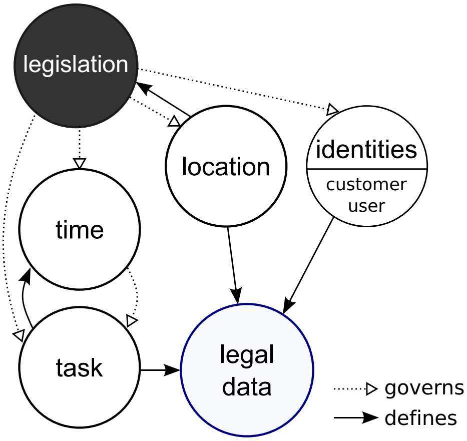


Fig. 2. Schema of a legislation government access control model

3.1 Modeling Legislation

Legislation defines a legal framework for daily business. Therefore, it governs every business action, every transaction and every data access. Figure 2 shows our used law model. Main component of the model is legislation of a region, country, or state. From the perspective of access control, legislation defines which data can be accessed legally with the given context. Laws govern the where (*location*), the who (*user*) and the whom (*customer*), the when (*time*) and the what (*task*). Contrariwise, the location defines, which legislation is valid and has to be observed. At the same time, legislation governs which location is permitted for a data access. Hence, location and legislation are mutually dependent. The identities of user and customer are also mutually dependent, as a user has various customers to advise and a customer may have various users as consultants. The relationship between both is defined in the law by a valid contract between both individuals, between an individual and a company/institution, or between two companies. Contracts including multiple parties are also possible. Time and task interact with each other. A task or an action that is legal is defined from the legislation. For this task, a specific point in time or a time range may be defined indirectly. Typically, the time range rather is specified by "as short as possible, as long as necessary" or during "working hours" than a specific time. Combining two or more sets of legislation will result in a set of legally accessible data for both sets of legislation with the given context. The various types of context information, which serve as basis for an law-aware access decision, remain the same. However, the result depends on the used sets of legislation.

3.2 Identities and Task

Identities (determined by *who* and *whom* in the context definition) are used in two ways: One identity identifies the user, e.g. all consultants of a company. This approach is well known for all access control systems. The other identity defines the customer who is the subject of the data.

A task describes what occurs in a specific situation (e.g. a customer advisory service). In our case, the notion task is a mandatory justification for mobile data access to sensitive data. However, a task is not necessarily required to access non-customer-related information or non-confidential data, such as product information. In our definition, a task is determined by an entry in the diary of a user that also may link the identities, data, location, and time.

3.3 Time

Time ranges represent a common access restriction. When a user requests access to specific information, the access control system checks whether the user is allowed to access this data at the current point in time. The context time is defined for customer-related data by a task. For access to non-customer-related data, the time range can be defined, for example, by a policy of the company to cover the itinerary working hours of a consultant. Due to the different time zones on a travel, the location context serves as input to calculate the correct local time. Sensitive data should be accessible for a limited time only to proactively minimize the risk of unauthorized access or data disclosure.

3.4 Location

Various approaches to context-aware access control systems [7,2] use a location as context information in decisions concerning access. Location describes the physical position of a mobile device. A position specifies not a single location point, but a location space. The method of determination defines the precision and the size of such location space. A position can be described by absolute values (GSM cell or GPS) or relative values (derived from an absolute position or from a proximity measurement).

In our approach, we distinguish between a *legislation location* and an *activity location*. The legislation location determines the validity area of a law – the country or region where a specific legislation needs to be observed. The activity location is the current location from where the user performs the access request. For the activity location, we differentiate between the *expected location* and the *current location*. An expected location is noted by the consultant in the diary. It specifies the location where the user will probably access the data, for example, at the location of the meeting with a customer. In our scenario, if a consultant has to travel abroad to accomplish a task, the supervisor gives an authorization for the travel in advance. This authorization confirms the expected location with a "second set of eyes". A current location is where a consultant is located during an access. This may be specified by an attribute of a subject. The OASIS organization designates the following attribute identifier for the current location of the subject: "urn:oasis:names:tc:xacml:3.0:ec-us:subject:location" [6].

We define a model called *zones*$^{+}$ to categorize locations. Zones^{+} is an XML location tree where a node can for example be a region (e.g. European Union), a country (e.g. the U.S.A, Japan or Germany) or a state (e.g. New York, Washington D.C.). The two children of such a node contain the areas separated into a restricted area where special law enforcement exist (e.g. in a customs duty area or a police station) and an unrestricted area, which contains all areas that are not defined as restricted. This bisection is used to support the insulation of sensitive data that should not be disclosed, for example, during a customs inspection where a consultant omitted to close the connection to a confidential resource at headquarters.

Policy 1. Definition of a legislation location within the target section

```
<Target>
  <Legislation>
  <LocationMatch MatchId="function:anyURI-equal">
    <AttributeValue DataType="string">Country A</AttributeValue>
    ...
    </LocationMatch>
  </Legislation>
</Target>
```

The definition of XACML itself provides no means of defining a legislative location. Thus, within the definition of the target an additional attribute extends the policy with an identifier that specifies the area where the policy is applicable (Policy 1). As the subject, resource, and action are checked, this attribute is included in the decision as to whether or not this policy is applicable to the current request. Therefore, if an access to data in country *S* is requested from another country *D*, both countries are checked. If the location defined in the policy equals either country *S* or *D* the policy becomes applicable unless subject, resource, and action do not match the request. Further on, a legislation tag can affiliate different countries. Provided that the countries concerned are listed as AttributeValues of a LocationMatch.

Remark that, in a standard XACML model the legislation-tag will be ignored and, therefore, the decision whether a policy applies to a request is based on subject, resource, and action only. A system, which does not evaluate the legislation tag, is rather over-restrictive than under-restrictive, as legislation-driven policies of different countries tend to constrain access. We also must point out, that instead of using the additional legislation tag to define a legislation location the property environment might be used. In our opinion, proceeding with this approach is more unambiguous than to merge the legislation location in an existing tag. However, this remains a subject to further discussions as the development of the XACML standard continues.

3.5 Legal Constraints

In the previous section we distinguished between *legislation location* and *activity location*. Legislation location means the validity area of a law, which we specify as a legal constraint. The *activity location*, which is the current position of the subject who requests the data, determines the legal constraints that have to be observed. A legislation location can be a *union* (e.g. European Union and the United States of America), a *country* (e.g. Germany, Japan, or Luxembourg), a *state* (e.g. Florida, California, or British Columbia) or an *organization* (e.g. Microsoft Corporation, Allianz SE), where the latter addresses organizational policies.

A single law can influence one or more of the other context information items, for example, if a law prohibits the use of strong encryption mechanisms, which another country presumes, a condition has to reflect this law. Additionally, laws can cause conditional constraints that relate to context information, but cannot be represented by one of the contexts described in this paper. Such a conditional constraint may be, for example, a signed customer agreement. In legal engineering two cases can occur. First, a policy is directly and unambiguously generated from the written natural form of an act. Secondly, a law cannot be transformed directly but has to be divided into several parts and has to be interpreted.

A Precedence-aware Policy-Combining Algorithm becomes necessary to handle the various levels of hierarchy that appear with laws of different controlling authorities. Depending on the controlling authority laws may overrule other laws, for example *national law* of a member state of the European Union overrules *European directives*.

Besides the *law* exists *precepts* (amongst others), which advise of a specific behavior or rule of action and is not mandatory. Existing XACML policy-combining algorithms do not take into account the priority of a law, which we call a precedence of a law. Algorithm 1 shows the pseudocode for a combining algorithm that respects the precedence of the sets of legislation. If the precedence of the first law (*law a*) is lower than the precedence of the second law (*law b*) the function is recalled with flipped variables, which does the implementation and, therefore, the cases to handle shorter (line 3). If both laws evaluate to indeterminate (Alg. 1, line 6) some error occurred and no decision can be made. Similarly, if laws have the same precedence but one evaluates to permit the other to deny, the total evaluation results in indeterminate (Alg. 1, line 10). This is,

Algorithm 1. Pseudocode of a Precedence-aware Policy-Combining Algorithm

```
 1: function $Combine_{law-permit-override}(a,b)$
 2:     if (Precedence(b) $>$ Precedence(a)) then
 3:         return $Combine_{law-permit-override}$ $(b,a)$     ▷ Re-call function with turned variables.
 4:     end if
 5:     if (a = indeterminate or b = indeterminate) then
 6:         return indeterminate     ▷ An error occurred and prevents a decision.
 7:     end if
 8:     if (Precedence(a) $=$ Precedence(b) then
 9:         if (a = deny and b = permit) or (a = permit and b = deny) then
10:             return indeterminate     ▷ Conflicting decision.
11:         end if
12:     else if (a = permit) then
13:         return permit     ▷ Both legislation allow access or law a overrules law b.
14:     else if (a = not applicable) then
15:         if (b = permit) then
16:             return permit     ▷ Only law b is applicable.
17:         else if (b = deny) then
18:             return deny     ▷ Law b denies access.
19:         else
20:             return not applicable     ▷ Both laws are not applicable.
21:         end if
22:     else
23:         return deny     ▷ For all remaining cases the access is denied
24:     end if
25: end function
```

however, a policy conflict that has to be resolved by an additional policy conflict handling. At the moment this has to be performed semi-automatic. Algorithm 1 respects the precedence, which means that as higher the number as “more” important is the law. In other words, if an organizational policy allows an access to data but a law with a higher precedence denies the same access, an access request has to be denied (Alg. 1, line 23). The combining algorithm denies by default (Alg.1, line 23).

It remains to mention that rules evaluating to “not applicable” neither permits nor denies.

This is different to “indeterminate” that indicates a problem that occurred during either the evaluation or the policy-combining process. The result indeterminate needs special attention. By default, indeterminate overrules a permit from another policy, which is required to keep data secure if a decision is not evaluated unambiguously.

Our proposed precedence-aware policy-combining algorithm is fully integrable into existing XACML systems to replace the default policy-combining algorithms.

4 Transformation

Transition of written legislation into a computer-useable form is an essential step preparing law-compliance of applications. As shown in Figure 3, taking legislation as input for a transformation is the first step. This transformation can either be performed automatically, semi-automatically or manually. In the latter case, lawyers and/or security officers read laws and interpret them. Mostly, they perform an interpretative translation and not a bijective transition of laws to policies. The documentation might refer to which policy reflects which law but is not necessarily mandatory. If a law changes the complete policy set needs to be revised in order to recover law-compliance. However, reassessing new or changed laws to existing policy sets is an error-prone and prolonged process if performed manually and includes more than one legislation, as it occurs for cross-border data access.

The output of the applying a transformation (see Fig. 3) is a single policy or set of policies. Such policies can be included into applications, application frameworks, or in any desired system that is able to interpret these policies. A verification process, which is denoted by (3) and (4) in Figure 3, shall cope with incompleteness or misinterpretations of the law-to-policy transformation process. Reporting missing legislative coverage Figure 3 (5) to the legislator to entail on future implementation of those missing laws completes the process of a law-to-policy transformation. Since laws are often written in a domain-specific, fuzzy, and stylistic-advanced notation steps (4) and (5) is

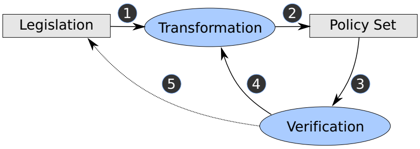


Fig. 3. Schema of a law to policy transition process

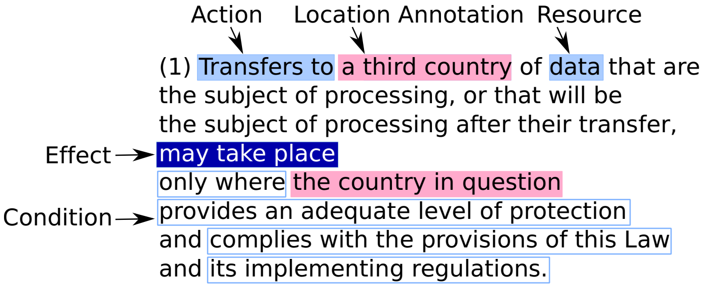
important for a transformation. Automatic transitions would require, however, a complete taxonomy of legal terms and their interrelations for an unambiguous interpretation.

Tanaka et al. presented in [12] a two-parted structure of law provisions of Japanese legislation, which consists of a *requisite part* (with subject and condition) and an *effectuation part* (with object, detail, and provision). This structure does also cover parts of the law sentences of the Luxembourgian law.

A single law can provide various types of information such as a: *a*) **condition**, which is a definition of exceptions or states for what a specific action is allowed or denied in a certain context. It defines what needs to be fulfilled to entitle an action; *b*) **default behavior** that defines the action if a decision cannot be made unambiguously; *c*) **detail**, which gives more information to specify a subject in more detail or to restrict effects, for example as annotation; *d*) **entitlement** that is a judgment-free statement of what right is granted; *e*) **general statement** specifying, for example, the purpose for which the law was defined; *f*) **link** as a reference to another law where, for example, a definition is refined or where exceptions may be specified; *g*) **nominal definition**, a definition of terms, which are used later within the document; *h*) **penalty**, a punishment, e.g. terms of imprisonment or payment of money, for breaking the specified law. Mostly, it defines a range from a minimum penalty to a maximum penalty.

To demonstrate the structure of laws, we use *Loi 02-08-2002* from Luxembourg, a law on the *Protection of Persons with regard to the Processing of Personal Data*. Figure 4 shows chapter IV, article 18 letter (1) – transfer of data to third countries of the Luxembourgian law for data protection and privacy. Article 18 describes the main principles.

As shown in Figure 4, letter (1) describes a resource (*data*), an action (*transfer to*) with an annotation of a location as detail (*a third country*), an effect (*may take place only*). The second part defines conditions for the effect (*only where*), which needs to comply to several provisions (*provides an adequate level of protection and complies with the provisions of this Law and its implementing regulations*). This annotation is currently performed by human experts. *A third country* attracts attention and on the first view all countries but Luxembourg are included. However, *Chapter 1, Art. 1:-Definitions, letter (m)* defines third countries as such that are not members of the European Union.



(1) Transfers to a third country of data that are
the subject of processing, or that will be
the subject of processing after their transfer,
may take place
only where the country in question
provides an adequate level of protection
and complies with the provisions of this Law
and its implementing regulations.

Fig. 4. Identifying transition properties in chapter IV, article 18, letter (1)

Policy 2. Target section of Loi 02-08-2002, chapter IV, article 18, letter 1

```
<Target>
 <!-- Applies to all Subjects -->
 <!-- Applies to all Resources -->
   <Actions>
     <Action>
       <ActionMatch
         MatchId="function:string-equal">
         <!-- Applies to the action transfer with a given destination -->
         <AttributeValue DataType="string">
           transfer
         </AttributeValue>
         <ActionAttributeDesignator AttributeId="action:action-id"
             DataType="string"/>
           <Apply FunctionId="function:not">
          <LocationMatch MatchId="function:string-equal">
    <AttributeValue DataType="string">
    destination
    </AttributeValue>
    <LocationMatch MatchId="function:destination-uri">
         <AttributeValue DataType="string">
         member of european union
    </AttributeValue>
    </LocationMatch>
     </LocationMatch>
    </Apply>
       </ActionMatch>
     </Action>
   </Actions>
   <!-- Applies to all Environments -->
 </Target>
```

Policy 2 shows an excerpt of the target section for chapter IV, article 18, letter (1) (see also Fig. 4). The policy is valid for any subject and any resource, which is designated by XACML through omitting the tags <*Subject*> and <*Resource*>. The policy uses, therefore, "transfer" as action in the target section including an additional location annotation that is expressed by a <*LocationMatch*> and a destination country that is not member of the European Union (<*Apply FunctionId="function:not"*>).

The part "complies with the provisions of this Law" is quite straight as our approach includes by default the legislation of source country and destination country of a connection (Policy 3). However, the specification in XACML is more complicated as the AttributeValue is not a static value but dynamic. In practice, the destination of a connection remains statically at one location, in our case Luxembourg. The source country of a connection may change between requests and is, therefore, dynamic. The first part "provides an adequate level of protection" appears very fuzzy. The law does not specify how protective measures are defined and how to satisfy those requirements. However, Chapter IV-Section 23 , *special security measures*, lists, for example, the protective measures of access control, usage control, transport control. To demonstrate how laws are we interpret and transform laws into policies we use, as example, *Loi 02-08-2002* from Luxembourg, a law on the *Protection of Persons with regard to the Processing of Personal Data*. Chapter IV, Section 5, letter (f) (Legitimacy of data processing) states that data processing is only permitted if the individual concerned has granted an agreement. Hence, it has to be checked whether a signed and valid agreement for this customer exist before deciding about access (Policy 4).

Policy 3. Condition to observe the legislation of source and destination of a connection

```
<Condition>
  <Apply FunctionId="setLegislation">
    <LegislationAttributeDesignator
      AttributeId="connectionSource" DataType="GEOLocation"/>
    <LegislationAttributeDesignator
      AttributeId="connectionDestination" DataType="GEOLocation"/>
  </Apply>
</Condition>
```

Policy 4. Condition for a signed customer agreement

```
<Condition FunctionId="function:not">
    <Apply FunctionId="urn:larbac:function:signedCustomerAgreement">
        <Apply FunctionId="function:boolean-equal">
            <AttributeValue DataType="string">true</AttributeValue>
        </Apply>
    </Apply>
</Condition>
```

This law is valid for any country, but needs only to be observed for customer-related data. Article 19. Derogations, letter (a) is written: “the data subject has given his consent to the proposed transfer”, which results in the same condition as Article 5, letter (f) (Policy 4).

5 Conclusion and Future Work

In this paper, we addressed the problem of using the eXtensible Access Control Markup Language (XACML) for law-aware access control. We stressed the necessity to incorporate legislation into mobile cross-border access. To this end, we demonstrated how various types of context information are defined within an XACML policy, and described the different types of context and their interrelations. We introduced a new policy-combining algorithm that respects precedences of laws, which became necessary to facilitate the evaluation of transformed policies of different controlling authorities.

Currently, we are implementing a prototype of the law-aware access control system using our law-enriched XACML policies. We are also investigating semi-automatic methods to transform laws into XACML policies that support human experts during a law to policy transformation. For this, we build up an ontology consisting of legal terminologies and a mapping to XACML policies.

Acknowledgement

This work was supported by the Fonds National de la Recherche Luxembourg under grant number AFR 05/109.

References

1. Bertino, E., Bettini, C., Ferrari, E., Samarati, P.: A Temporal Access Control Mechanism for Database Systems. IEEE Transactions on Knowledge and Data Engineering 8(1), 67–80 (1996)

2. Damiani, M.L., Bertino, E., Catania, B., Perlasca, P.: GEO-RBAC: A Spatially Aware RBAC. ACM Trans. Inf. Syst. Secur. 10(1), 2 (2007)
3. Dey, A.K., Abowd, G.D.: Towards a Better Understanding of Context and Context-Awareness. In: Computer Human Intraction 2000 Workshop on the What, Who, Where (1999)
4. Katayama, T.: Legal Engineering - An Engineering Approach to Laws in e-Society Age. In: Proceedings of the 1st International Workshop on JURISIN (2007)
5. Moses, T.: eXtensible Access Control Markup Language TC v2.0 (XACML). In: Organization for the Advancement of Structured Information Standards (OASIS) (February 2005)
6. Organization for the Advancement of Structured Information Standards (OASIS). XACML 3.0 Export Compliance-US (EC-US) Profile Version 1.0 (September 2009)
7. Schilit, B., Adams, N., Want, R.: Context-Aware Computing Applications. In: IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, US (1994)
8. Serban, C., Chen, Y., Zhang, W., Minsky, N.: The Concept of Decentralized and Secure Electronic Marketplace. Electronic Commerce Research 8(1-2), 79–101 (2008)
9. Stieghahn, M., Engel, T.: Law-aware Access Control for International Financial Environments. In: MobiDE 2009: Proceedings of the Eighth ACM International Workshop on Data Engineering for Wireless and Mobile Access, pp. 33–40. ACM, New York (2009)
10. Stieghahn, M., Engel, T.: Using XACML for Law-aware Access Control. In: 3rd. International Workshop on Juris-informatics (JURISIN), pp. 118–129 (2009)
11. Strembeck, M., Neumann, G.: An Integrated Approach to Engineer and Enforce Context Constraints in RBAC Environments. ACM Trans. Inf. Syst. Secur. 7(3), 392–427 (2004)
12. Tanaka, K., Kawazoe, I., Narita, H.: Standard structure of legal provisions - for the legal knowledge processing by natural language (in Japanese). IPSJ Research Report on Natural Language Processing, 79–86 (1993)
13. Ungureanu, V., Minsky, N.H.: Establishing Business Rules for Inter-Enterprise Electronic Commerce. In: Herlihy, M.P. (ed.) DISC 2000. LNCS, vol. 1914, pp. 179–193. Springer, Heidelberg (2000)