IADIS International Journal on Computer Science and Information Systems
Vol. 9, No. 1, pp. 16-29
ISSN: 1646-3692

# CONFLICTS BETWEEN GEOXACML ACCESS CONTROL POLICIES IN GEOGRAPHIC INFORMATION SYSTEMS

Mohamed Yahiaoui. *Faculty of sciences FSDM, USMBA University, P.O.B 1796 Atlas, Fez, Morocco. yahiaoui.med@gmail.com*

Ahmed Zinedine. *Faculty of sciences FSDM, USMBA University, P.O.B 1796 Atlas, Fez, Morocco. ahmedzinedine@yahoo.com*

Mostafa Harti. *Faculty of sciences FSDM, USMBA University, P.O.B 1796 Atlas, Fez, Morocco. mharti@rocketmail.com*

**ABSTRACT**

In our days, geospatial data play a key role in a wide spectrum of critical fields, such as disaster and emergency management, environmental monitoring, land and city planning. The specific nature of these geospatial data presents many challenges in terms of security. Indeed, security issues for geospatial data are different and in many ways more complex than security issues for relational data. Among these challenges, we find the detection and resolution of conflicts/anomalies between access control policies which is a very common situation in this kind of systems.
In this work we deal with the problem of detection and resolution of conflicts/anomalies between GEOXACML policies. That is the case where several policies provide conflicting answers to a same request. We give more attention to the mathematical formalism of the problem. We introduce the notion of the *canonical representation* of the query space which is a classification that provides a natural way to handle interferences between policy targets (in other words conflicts/anomalies). Then we highlight the key role that this canonical representation will play as the core of a future Framework for conflict detection and resolution in GEOXACML access control systems.

**KEYWORDS**

Geographic Information Systems; Access Control Policies; GeoXACML; Conflict Detection and Resolution; Canonical Representation.

## 1. INTRODUCTION

The GeoXACML standard is a geospatial extension based on extensibility points of XACML (*eXtensible Access Control Markup Language)*. It allows the integration of the geographic information specificities in the access control systems. In particular, it allows the use of spatial attributes in the declaration and enforcement of restrictions in access control. In distributed GIS environments, GeoXACML allows a combined use of the geographic information provided by multiple organizations. GeoXACML provides a common language for specifying, sharing and enforcing spatial data access rights [19].
One of the biggest challenges that face GeoXACML-based access control systems is the difficulty to ensure consistency between the different policies. Indeed, it is very common in the context of GIS that multiple authorities intervene independently in the specification of policies, which increases the risk of anomalies and conflicts (in the sense that several policies can give conflicting answers to a same access request).

The overall policy consistency check is the responsibility of the access control administrators. The complexity of geo-spatial data and systems makes difficult -if not impossible- manual detection of anomalies and conflicts between access control policies. Therefore, the use of specialized tools is necessary. But till now, GeoXACML suffers from the absence of such tools.

The issue of detecting and resolving conflicts between access control policies drew a lot of attention recently. Several approaches have been developed to address this problem.

In [6], the authors raised the issue of conflicts/anomalies between policies controlling access to geospatial data as a problem related to policy integration. They raised a scenario that highlighted the importance and also the complexity of the detection and resolution of conflicts/anomalies in the geographical context.

The paper [13] introduced an approach to analyze the Web access control policies expressed in XACML. This approach is based on a segmentation technique to identify policy anomalies and derive resolutions based on the standard combination algorithms.

The article [10] presents a software named "Margrave" used for role-based access control policies analysis. The idea is to translate XACML polices to Margrave own schema, and then make them available to users in order to run queries. The goal is to monitor and analyze the overall behavior of the access control system. According to the authors, "Margrave" can be used by designers, developers and administrators of access control systems.

The paper [14] proposed a unified Framework for policy analysis, detection and resolution of anomalies. The Framework is based on a generic approach to capture the semantics of common policies.

The authors of [15] introduced a formal model consistent with the Alloy language for the specification of access control policies. They used this model for automatic detection of anomalies using the Alloy analyzer.

For our part, we have addressed this issue in the case of XACML policies. In [25] we established a mathematical formalism of conflict between XACML-based access control policies. This formalism allows a comprehensive representation of all possible conflict situations. Then, in [26] we proposed a comprehensive Framework of detection and resolution of conflicts between XACML access control policies. This framework is based on the mathematical formalism that we have developed in [25]. The aim was to provide policy administrators with a tool for decision-making support. In fact, the Framework can be used to analyze the impact of policy edition (adding, modifying and deleting policies) before its integration into a production environment.

In this paper, we extend these results to the case of geographic information systems. The complexity of these systems and their special structure and organization make the detection and resolution of conflicts between policies a very complex and difficult task. For example, any system controlling access to geospatial data must take into consideration the following points:

- Layers and objects covered by the access control are often in interference,
- Possibility of multiple representations of a same object in different layers,
- Support for the geographical aspect of data,
- Multiple involved authorities.

Our goal is to provide systems controlling access to geospatial data with conflict detection mechanisms and tools that can handle the complexity of the structure and organization specific to geographic information systems.

In the proposed approach, we express the notion of conflict in a formal way by introducing a canonical representation of the query space as the quotient space of a given equivalence relation. That is a segmentation of the query space into subsets (authorization classes). Each class can be seen as the target of a policy (in a future conflict-free system). We then calculate the decision of this policy by combining different active policies on each authorization class. To this end, we use policy integration expressions introduced by the FIA algebra [23]. These expressions provide a mechanism richer than the standard GeoXACML combining algorithms. Then we give a method for calculating the projection of the FIA policy integration expression on the different authorization classes. This allows bringing the study of conflicts from the query space to its canonical representation.

At the end, we study the impact of the dynamic aspect of the policy repository. So we study the evolution of the canonical representation of the query space following the addition, modification and deletion of policies from the policy repository. In fact, this dynamic aspect of the policy repository is one of the concerns of the access control system administration that requires suitable decision-making support. Hence the importance of integrating the canonical representation as a part of a Framework for conflict detection and resolution in GeoXACML access control systems.

# 2. PRELIMINARIES

## 2.1 Access control in geographic information systems

Geographic information systems (GIS) provide a computing platform consisting of hardware, software, and processes designed to facilitate collecting, sharing, manipulating, analyzing, modeling and displaying geospatial data. These systems represent the spatial environment by using graphic primitives, such as points, arcs, polygons or raster. They combine these primitives with qualitative information such as the nature (city, building, road, forest, etc…) or any other contextual information (population, type, or a collectivity surface area).
Geospatial data have an effective presence in a wide variety of applications, for example we cite applications for disaster and emergency management, environmental monitoring, etc…, which often require coordination between various organizations, their data repositories, and users with different responsibilities [6].
The issue of security of geospatial data is one of the most important concerns in the field of GIS. Indeed, the proliferation of geospatial applications, the critical nature of data, the variety of application areas in addition to the multiplicity of stakeholder, increase the importance of the issue.
Security issues for geospatial data are different and in many ways more complex than security issues for relational data. These differences concern both the data organization and structures, and in particular the ways the data are used. In a GIS, data is typically organized in different thematic layers; these layers, which can be large in number, represent different aspects of an application domain [6].
Any modern access control system should be able to express:

- Fine-grained, positive and negative access rules;
- Content dependent access rules;
- Context dependent access rules.

In geographic information systems (GIS), request access to geospatial data may be made on the basis of the attributes, data type, content and context. Moreover, it is necessary to express access control rules based on spatial attributes.
GeoXACML allows the declaration and interoperable enforcement of access restrictions to protect geospatial data. The OGC has standardized the use of GeoXACML to endow the GIS systems with interoperable access control that supports the combined use of protected and distributed geospatial-data.

## 2.2 GEOXACML

The GeoXACML standard is a geospatial extension of the eXtensible Access Control Markup Language (XACML), the OASIS standard for the implementation of access control systems [22]. The aim is to use the XACML extensibility points in order to integrate the declaration and enforcement of access restrictions based on spatial attributes.
GeoXACML provides support for spatial data types and spatial authorization decision functions. Those data types and functions can be used to define additional spatial constraints for XACML based policies [19].
In short, GeoXACML defines:

- The geometry model on which the geometric data types in access rules have to be based on;
- The different encoding languages for geometric data types;
- The testing functions for topological relationships between geometries; and
- The geometric functions.

### 2.2.1 Policy

The access control policies are used to define system behavior against access requests. Indeed, a policy is a set of rules on the attributes of subjects, resources, actions and environment. These rules are combined using logical operators to determine the rights of users on system resources.

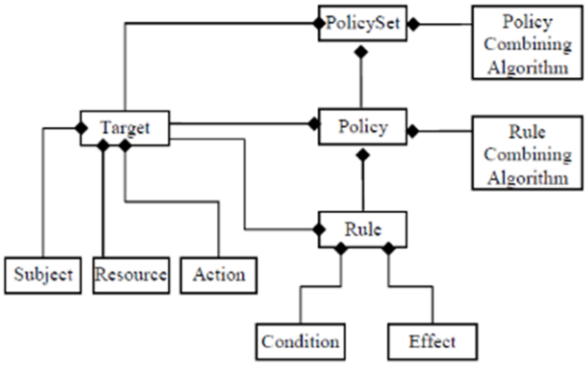


Figure 1. GeoXACML Policy language model [22]

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy  xmlns=".." xmlns:gml=http://www.opengis.net/gml PolicyId="urn:oasis:names:tc:example:WMSC23"
            RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
      <Target />
      <Rule Effect="Permit" RuleId="test_1.0.1">
            <Target>
                  …
            </Target>
            <Condition>
                  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
                        …
                        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:all-of">
                              <Function FunctionId="urn:ogc:def:function:geoxacml:1.0:geometry-contains" />
                              <AttributeValue DataType="urn:ogc:def:dataType:geoxacml:1.0:geometry">
                                    <gml:Polygon gid="Europe" srsName=" http://www.opengis.net/gml/srs/epsg.xml#4326"
                                                xmlns:gml="http://www.opengis.net/gml">
                                          <gml:outerboundaryIs>
                                                <gml:LinearRing>
                                                      <gml:coordinates cs="," ts=" ">
                                                      -11,55 -10,35 -5.5,36 -1,36 1,38 5,38 11 38 14 36 26 33 29,36 26,39
                                                       29,46 39 47 40 49 27,56 27,60 25,60 20,58 21,56 19,55 11,55 10,57
                                                       7,57 8,54 3,53 -2,60 -8,58 -11,55 -11,55
                                                      </gml:coordinates>
                                                </gml:LinearRing>
                                          </gml:outerboundaryIs>
                                    </gml:Polygon>
                              </AttributeValue>
                              <AttributeSelector RequestContextPath="//gml:boundedBy/gml:LinearRing"
                                                DataType="urn:ogc:def:dataType:geoxacml:1.0:geometry" />
                        </Apply>
                  </Apply>
            </Condition>
      </Rule>
</Policy>
```

Figure 2. A simplified example of GeoXACML Policy.

### 2.2.2 Request

GeoXACML requests are written in an XML encoding that allows expressing the attributes of the user, the requested resource, the desired action and other environmental information. The requests are sent to the PDP (see next section). This last extracts the attributes of the request and compares them with the targets of policies to determine GeoXACML policies that are applicable to this request and then determines the decision.

The GeoXACML policy language defines three different decisions: (i) Permit, (ii) Deny, (iii) Not applicable.

```
<?xml version="1.0" encoding="UTF-8"?>
<Request ...>
```

```
<Subject>
        <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
                <AttributeValue>00113</AttributeValue>
        </Attribute>
</Subject>
<Resource>
        <ResourceContent>
                <gml:boundedBy>
                        <gml:Box srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
                                <gml:coord>
                                        <gml:X>-97.105</gml:X><gml:Y>24.913</gml:Y>
                                </gml:coord>
                                <gml:coord>
                                        <gml:X>-78.794</gml:X><gml:Y>36.358</gml:Y>
                                </gml:coord>
                        </gml:Box>
                </gml:boundedBy>
        </ResourceContent>
        <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
                DataType="http://www.w3.org/2001/XMLSchema#string">
                <AttributeValue>01-test</AttributeValue>
        </Attribute>
</Resource>
<Action>
        <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
                DataType="http://www.w3.org/2001/XMLSchema#string">
                <AttributeValue>map</AttributeValue>
        </Attribute>
</Action>
</Request>
```

Figure 3. A simplified example of GeoXACML Request.

### 2.2.3 GeoXACML Data-flow Model

The GeoXACML data-flow model defines a modular and distributed architecture of the access control system. It defines the various components and their roles. It defines also the possible exchanges of messages between these components and the structure of these messages. The figure below illustrates this architecture.

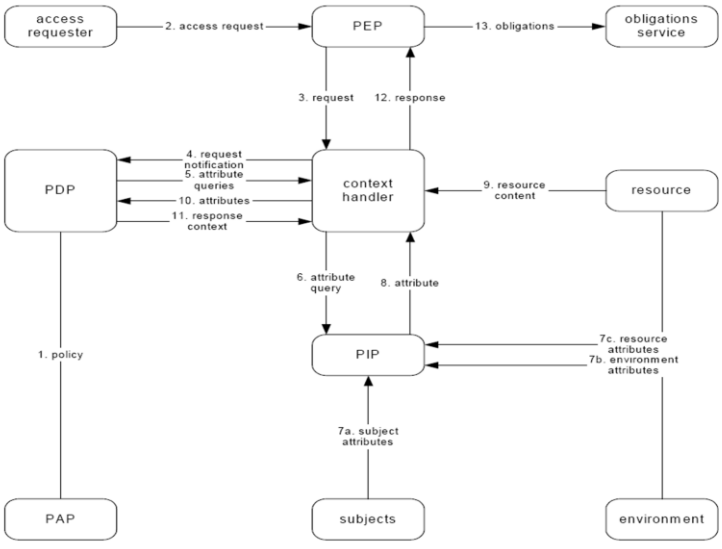


Figure 4. GeoXACML data-flow diagram [22]

PAP: The Policy Administration Point is the point of administration of policies. It allows administrators to maintain the policy repository.

PEP: The Policy Enforcement Point is the component that receives access requests to resources. It is responsible for transforming these requests to GeoXACML format, and then it transmits them to the PDP. Depending on the response of the PDP, the PEP allows or denies access to the resource. In case of error a feedback can be raised.
PDP: The Policy Decision Point is the component that handles GeoXACML requests received from the PEP and then gives the appropriate GeoXACML response. This is an authorization decision (Permit/Deny) based on the policies stored in the PAP.

## 2.3 The Fine-Grained Integration Algebra (FIA)

The GeoXACML standard policy combining algorithms allow the implementation of strategies for policy integration. But they do not give enough flexibility to support the requirements of applications and involved parties.
To resolve this problem the FIA algebra was introduced in [23]. It provides an efficient mechanism for policy integration. It can support a strategy of flexible and granular policy integration. Among the advantages of the FIA algebra we quote the following:

- It supports policies expressed in rich languages like GeoXACML;
- It allows to solve the problems related to the integration of heterogeneous and fine-grained access control policies;
- This algebra is able to support the specification of a wide variety of integration constraints;
- The algebra is highly expressive.
- We give below a quick overview of the FIA algebra and its main concepts.

### 2.3.1 The semantics of queries and policies

The formalism of queries is based on the finite set '*A*' of attribute names. *A* is composed of attributes characterizing the subjects, resources, actions and environment. Each element of '*A*' has a domain, denoted dom(*a*), consisting of all possible values of the attribute *a*.
***Definition 1*** A request r is defined as follows: $r \equiv \{(a_1, v_1), (a_2, v_2), \cdots, (a_k, v_k)\}$, where $a_1, a_2, ..., a_k$ are attribute names, and $v_i \in dom(a_i)$ $(1 \leq i \leq k)$.
***Definition 2*** let P be an access control policy. The semantics of P is defined as a 2-uplet $\langle R_Y^P, R_N^P \rangle$, where $R_Y^P$ (resp. $R_N^P$) is the set of allowed (resp. denied) requests. Note that $R_Y^P \cap R_N^P = \emptyset$.

### 2.3.2 The operators of the FIA algebra

The fine-grained integration algebra (FIA) is given by $\langle \sum, P_Y, P_N, +, \&, \neg, \prod_{dc} \rangle$, where $\sum$ is a vocabulary of attribute names and their domains, $P_Y$ and $P_N$ two policy constants, + and & are two binary operators, and ¬ and $\prod_{dc}$ are two unary ones.
To illustrate the operators and constants of the FIA algebra let $P_1$, $P_2$ and $P_I$ be three policies such that $P_1 \equiv \langle R_Y^{P1}, R_N^{P1} \rangle$, $P_2 \equiv \langle R_Y^{P2}, R_N^{P2} \rangle$ and $P_I \equiv \langle R_Y^{PI}, R_N^{PI} \rangle$.
***Permit policy ($P_Y$)***. $P_Y$ is a policy constant that permits everything.
***Deny policy ($P_N$)***. $P_N$ is a policy constant that denies everything.
***Addition (+)*** : the addition of two Policies P1 and P2 is a policy that allows all requests authorized by P1 or P2, and refuses requests that are denied by one of the two policies and not authorized by the other. More precisely: $P_I = P_1 + P_2 \Leftrightarrow R_Y^{PI} = R_Y^{P1} \cup R_Y^{P2} \wedge R_N^{PI} = (R_N^{P1} \backslash R_Y^{P2}) \cup (R_N^{P2} \backslash R_Y^{P1})$
***Intersection (&):*** $P_I = P_1 \,\&\, P_2 \Leftrightarrow R_Y^{PI} = R_Y^{P1} \cap R_Y^{P2} \wedge R_N^{PI} = R_N^{P1} \cap R_N^{P2}$
***Negation (¬):*** $P_I = \neg P \Leftrightarrow R_Y^{PI} = R_N^P \wedge R_N^{PI} = R_Y^P$.
***Domain projection ($\prod_{dc}$):***
***Definition 3*** A domain constraint dc takes the form $\{(a_1, plage_1), (a_2, range_2), \ldots, (a_k, range_k)\}$, where $a_1, a_2, \ldots a_k$ are attribute names, and $range_i$ $(1 \leq i \leq k)$ are sets of values from the vocabulary $\sum$. Given a request $r = \{(a_{r1}, v_{r1}), (a_{r2}, v_{r2}), \cdots, (a_{rm}, v_{rm})\}$. We say that r satisfies dc if the following condition holds: for each $(a_{rj}, v_{rj}) \in r$ $(1 \leq j \leq m)$, if there exists $(a_{rj}, range_i) \in dc$, then $v_{rj} \in range_i$.
The semantics of $(\prod_{dc})$ is given by
$P_I = \prod_{dc}(P) \Leftrightarrow R_Y^{PI} = \{r \mid r \in R_Y^P$ and r satisfies dc $\}$, $R_N^{PI} = \{r \mid r \in R_N^P$ and r satisfies dc $\}$.

***Not-applicable policy ($P_{NA}$)*** $P_{NA}$ is a policy constant that is not applicable for every request. It is defined as $P_{NA} = P_Y \& P_N$.
***Effect projection ($\Pi_Y$ and $\Pi_N$)*** $\Pi_Y$ (P) restricts the policy P to the requests allowed by it. It is defined as: $\Pi_Y$ (P) = P $\& P_Y$. Similarly, $\Pi_N$(P) restricts the policy P to the requests denied by it; it is defined as $\Pi_N$(P) = P $\& P_N$.
***Subtraction (−)*** Given two policies $P_1$ and $P_2$. The subtraction operator is defined as: $P_1 - P_2 = (P_Y \& (\neg(\neg P_1 + P_2 + \neg P_2))) + (PN \& (P_1 + P_2 + \neg P_2))$.
***Precedence ($\triangleright$)***: Given two policies $P_1$ and $P_2$, $P_1 \triangleright P_2$ is the policy that gives the same decision as $P_1$ for all queries applicable to $P_1$ and gives the same decision as $P_2$ for all other queries. It can be easily shown that $P_1 \triangleright P_2 = P_1 + (P_2 - P_1)$.

### 2.3.3 The expressions of the FIA algebra

A FIA expression is recursively defined as follows:

- If P is a policy, then P is a FIA expression.
- If f1 and f2 are FIA expressions so are (f1) + (f2), (f1)&(f2), and ¬(f1).
- If f is a FIA expression and dc is a domain constraint then $\Pi_{dc}$(f) is a FIA expression.

The FIA algebra can express not only the standard algorithms for combining XACML policies, but also other more complex integration constraints. We give below some examples:
Let $P_1$, $P_2$,..., $P_n$ be access control policies. The standard algorithms permit-overrides, *Deny-overrides*, first-one-applicable and only-one-applicable can be expressed respectively by the following FIA expressions: $P_1+P_2+ ... +P_n$, $\neg((\neg P_1) + (\neg P_2) + ... + (\neg P_n))$, $P_1 \triangleright P_2 \triangleright \ldots \triangleright P_n$ and $(P_1 - P_2 - P_3 - \ldots - P_n) + (P_2 - P_1 - P_3 - \ldots - P_n) + \ldots + (P_n - P_1 - P_2 - \ldots - P_{n-1})$.
Now we give an example of a more specific and more complex integration constraint. Let $P_1$, $P_2$ and $P_3$ be three policies. We aim to express the following integration constraint: For a given query $r_1$, if $P_1$ allows $r_1$ then the final decision will be that of the policy $P_2$, if not the final decision will be that of $P_3$. This integration constraint can be expressed by the following FIA expression:
$\Pi_Y(P_1 \& P_2) + \Pi_N(\neg P_1 \& P_2) + \Pi_Y(\neg P_1 \& P_3) + \Pi_N(P_1 \& P_3))$

## 2.4 Representation of GeoXACML policies by compound Boolean expressions over request attributes

GeoXACML policies can be transformed into compound Boolean expressions over request attributes where each compound Boolean expression consists of atomic Boolean expressions (AE) combined using the logical operations "∨" and "∧". Atomic Boolean expressions that appear in most policies fall into one of two categories:
i. Constraints of equality to a constant: $a = c$, $a \neq c$, where $a$ is an attribute name and $c$ is a constant.
ii. Constraints of inequality to a constant: $c_1$ $rel_1$ a $rel_2$ $c_2$, where $a$ is an attribute name, $c_1$ and $c_2$ are two constants and $rel_1$, $rel_2 \in \{<, \leq, >, \geq\}$.
So a policy can be expressed as follows:

$$P(x) = \begin{cases} Y, \ if \ f_1(x_1, x_2, \ldots, x_n) \\ N, if \ f_2(x_1, x_2, \ldots, x_n) \end{cases}$$

Where $x$ is a query and $x_1$, $x_2$ … $x_n$ are the values of attribute names of the query x and $f_1$ and $f_2$ are two Boolean expressions.

# 3. MOTIVATION EXAMPLE

We illustrate the issues of conflicts/anomalies between policies through an example in a geographic information system of a city that is open to General Public and Town Planners groups. The system consists of three layers: buildings, roads and population. The access control requirements are as follows:

- The General Public group has read access to the road layer.

- The group Town Planners accesses all levels for reading and writing.

The implementation of these requirements is made by the following policies:

$P_1$: (Role = General Public, resource = all features of road layer, action = read, decision = Permit)
(Role = General Public, resource = all features of all layers, action = write, decision = Deny)

$P_2$: (Role = Town Planners, resource = all features of all layers, action = read/write, decision = Permit)

Suppose now that the administrator of access control system decides to ban access to features (of all layers) located near a sensitive infrastructure (the aim is not to reveal the presence of this last). This requirement can be implemented via the following policy:

$P_3$: (Resources = all features of all layers, feature.geom $\cap$ $polygon_1 \neq \phi$, decision = Deny), where Polygon1 is a polygon containing the sensitive infrastructure.

It is obvious that $P_3$ is in conflicts/anomalies with existing policies.

To eliminate these conflicts/anomalies, the administrator of the access control system has two solutions:

i. Use of standard policy combining algorithms.

ii. Redesign of all existing policies to build a system without conflicts/anomalies.

It is clear that the last solution is not practical, because of existing policies may be quite numerous and complex to the point where the redesign is almost impossible. While the first option, the following question is obvious: what policy combining algorithm should be used to meet the above requirements? The following example shows that, sometimes, none of the standard combining algorithms produces the intended result.

## 3.1 Case study

Let $Polygon_2$ be another polygon such that $Polygon_2 \cap Polygon_1 = \phi$. Consider the following requests:

$R_1$: (role = General Public, resource = road layer, action = read, extent = $Polygon_2$). It is clear that this request will be intercepted only by policy $P_1$.

$R_2$: (role = Town Planners, resource = building layer, action = read, layer extent = $Polygon_2$). This request will be intercepted only by policy $P_2$.

$R_3$: (role = General Public, resource = population layer, action = read, extent = $Polygon_1$). This request will be intercepted only by policy $P_3$.

$R_4$: (role = Town Planners, resource = road layer, action = write, extent = $Polygon_2$). This request will be intercepted by policies $P_1$ and $P_2$,

$R_5$: (role = General Public, resource = road layer, action = read, extent = $polygon_1$). It is clear that this request will be intercepted by policies $P_1$ and $P_3$,

$R_6$: (role = Town Planners, resource = building layer, extent = $Polygon_1$). This request will be intercepted by policies $P_2$ and $P_3$,

$R_7$: (role = Town Planners, resource = road layer, action = read, extent = $Polygon_1$) It is clear that this request will be intercepted by policies $P_1$, $P_2$ and $P_3$.

| R | $P_1$ | $P_2$ | $P_3$ | intended Result | permit-overrides | Deny-overrides | first-one-applicable | only-one-applicable |
|---|---|---|---|---|---|---|---|---|
| $R_1$ | Y | NA | NA | Y | Y | Y | Y | Y |
| $R_2$ | NA | Y | NA | Y | Y | Y | Y | Y |
| $R_3$ | NA | NA | N | N | N | N | NA | N |
| $R_4$ | N | Y | NA | Y | Y | N | N | Y |
| $R_5$ | Y | NA | N | N | Y | N | Y | NA |
| $R_6$ | NA | Y | N | N | Y | N | NA | NA |
| $R_7$ | Y | Y | N | N | Y | N | Y | NA |

The previous table shows that none of the traditional combining algorithms can integrate all three policies by producing the intended decisions that meets the previously specified requirements. Indeed each request in the previous example reflects a particular conflict situation. Each situation is characterized by the set of policies applicable to the request. But the standard combining algorithms lack the necessary mechanisms to customize the resolution of conflict.

In order to build an efficient solution for dealing with conflicts/anomalies, we include some requirements derived from the above example:

- A method of segmenting the query space into zones of conflicts/anomalies, able to give us an advanced understanding of policy interference.
- Tools and algorithms able to identify policy interference zones by a set of conditions on the attributes of subjects, resources and environment.
- The ability to use more efficient algorithms for policy integration.

The solution we propose to meet this needs is to implement a Framework for detection and resolution of anomalies. This framework must meet the following specifications:

- The Framework will be an extension of the PAP.
- It automates the detection of conflicts/anomalies and provide a clear representation,
- It updates the list of conflicts/anomalies after the addition, deletion and modification of policy,
- It Provides automatic resolutions based on customized combination algorithms (expressed using FIA expressions),
- It provides the possibility for administrators to manually implement the resolutions,

It makes available to the PDP an error-free representation of the set of policies. This representation must be updated automatically following addition or deletion of policies.

# 4. DETECTION AND RESOLUTION OF CONFLICTS/ANOMALIES BETWEEN GEOXACML POLICIES

## 4.1 Definitions and mathematical formalism

***Definition 4*** [23] (*Authorization Space*) let P be a policy. The authorization space of P is defined as the set of requests $Q_p$ for which the policy P is applicable, i.e. $Q_P = R_Y^P \cup R_N^P$. It can be given by:
$Q_p = \{x \in R_\Sigma / f(x_1, x_2, \dots, x_n)\}$, where $f = f_1 \vee f_2$ is the Boolean expression that defines P and $x_1$, $x_2$ … $x_n$ are the attributes of x.
An anomaly is defined as a situation where several policies intercept a same query to give access decisions (allow / deny). If, moreover, these decisions are contradictory, this is said to be a conflict. This situation is interpreted by the intersection of several authorization spaces.
***Definition 5*** (*conflict/anomaly*) let $P_1$ and $P_2$ be two policies. We say that there is a conflict/anomaly between $P_1$ and $P_2$ if $Qp_1 \cap Qp_2$ is not empty.
In the following we propose a method to classify queries so that derived classes include queries that can be processed by the system in similar ways. A class corresponds to the queries intercepted by exactly the same policies. Then we give the following definition of an authorization class:
***Definition 6*** (*authorization class*) let $E' = \{P'_1, \dots, P'_m\}$ be a given subset of policies. The authorization class of E' is defined as the set of all requests to which $P'_1, \dots, P'_m$ -and only $P'_1, \dots, P'_m$- are applicable. We note this set *AC(E')*.

## 4.2 The canonical representation of the query space

Let E be a finite set of policies, $\sum$ is a vocabulary of attribute names and their domains. The query Space $R_\Sigma$ has a canonical representation that reflects the behavior of policies against requests. We introduce below this canonical representation.
Let $r_1$ and $r_2$ be two requests of $R_\Sigma$. We denote by $\Re$ the relationship in $R_\Sigma$ defined by $r_1 \Re r_2$ if, and only if, for each policy P in E, $r_1 \in Q_P \Leftrightarrow r_2 \in Q_P$. In other words, the set of applicable policies to $r_1$ is exactly the set of those applicable to $r_2$.
One can easily show that the relationship «$\Re$» is an equivalence relation. We denote by $\overline{r^E}$ the equivalence class of the request r relatively to E.
***Definition 7*** The quotient space $R_\Sigma/\Re$ that is constituted of all equivalence classes is called *the canonical representation* of $R_\Sigma$.
Since the set E is finite, the set $\wp(E)$ of subsets of E is also finite. And since by definition each equivalence class $\overline{r^E}$ is associated with an element of $\wp(E)$ (we denote it $P(\overline{r^E})$) then the set of all equivalence classes is

finite. Therefore there exist $r_1, \ldots, r_n \in R_\Sigma$ such that $(\overline{r_i^E})_{(1 \le i \le n)}$ is a partition of $R_\Sigma$. From the well-known properties of equivalence relations this partition is unique. Hence the following result:

***Proposition 1*** There exists a finite set of requests $r_1, \ldots, r_n \in R_\Sigma$ such that $(\overline{r_i^E})_{(1 \le i \le n)}$ constitute the canonical representation of $R_\Sigma$.

Let $E' = \{P'_1, \ldots, P'_m\}$ be a subset of the set E of all policies in the access control system. Then for each request r in AC(E') (see definition 6) we have $AC(E') = \overline{r^E}$.

The following result allows calculating, more explicitly, authorization classes using Boolean expressions.

***Proposition 2*** Let $E = \{P_1, P_2, \ldots, P_n\}$ be the set of all policies in the access control system. For each P in E denote $f_P$ the Boolean expression defining the authorization space of the policy P (see [23]) and let $P(\overline{r^E}) = b$. One can easily show that:

$$\overline{r^E} = \{r \in R_\Sigma / (\bigwedge_{P \in b} f_P(x_1, x_2, \ldots, x_m)) \wedge (\bigwedge_{p \in b^c} \neg f_P(x_1, x_2, \ldots, x_m))\}$$

## 4.3 Illustration example

Let $E = \{P_1, P_2, P_3, P_4\}$, where $P_1$, $P_2$, $P_3$ and $P_4$ are GeoXACML policies and r is a request.

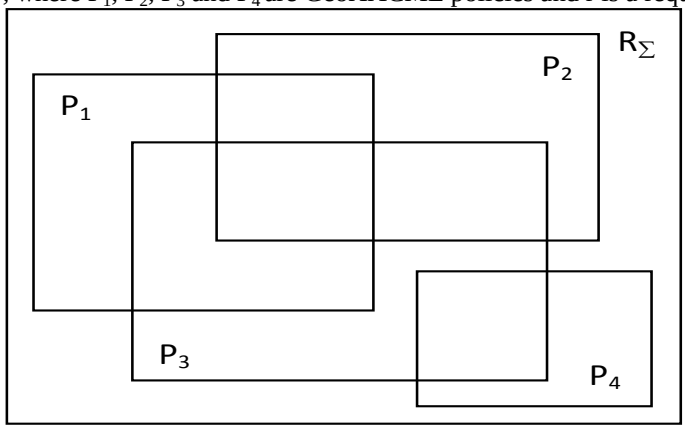


Figure 5. illustration of the authorization spaces in $R_\Sigma$

In the figure above $R_\Sigma$ represents the query space. Each rectangle represents requests that are intercepted by a given policy. For example the rectangle $P_1$ surrounds all requests to which the policy $P_1$ applies. Note that the request r is intercepted by $P_1$ and $P_3$ (and only by $P_1$ and $P_3$).

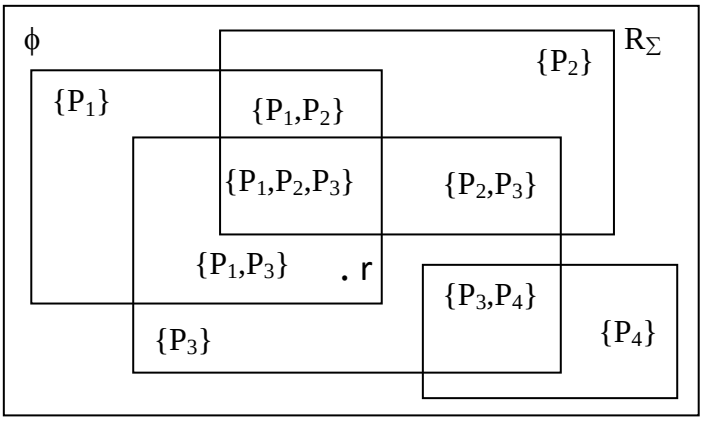


Figure 6. illustration of the canonical representation of $R_\Sigma$

In the figure above $R_\Sigma$ is divided into 10 zones (deduced from Figure 5): $\{P_1\}$, $\{P_2\}$, $\{P_3\}$, $\{P_4\}$, $\{P_1,P_2\}$, $\{P_1,P_3\}$, $\{P_2, P_3\}$, $\{ P_3,P_4\}$, $\{P_1,P_2,P_3 \}$ and $\phi$.
Each zone represents all requests intercepted exactly by a given set of policies. For example the zone $\{P_1, P_3\}$ represents requests intercepted by, and only by, $P_1$, $P_3$. $\{P_4\}$ represents requests intercepted by, and only by, $P_4$. The zone denoted $\phi$ represents queries that do not match any policy.
Indeed, each zone corresponds to an equivalence class of the canonical representation of $R_\Sigma$. In the example above the zone $\{P_1,P_3\}$ is the equivalence class of r.
We note that these classes correspond to possible interferences between the policies of E. We also note that.

- By definition each authorization class $\overline{r^E}$ is generated by an element of $\wp(E)$. Denote this element $P(\overline{r^E})$. (in the previous figure $P(\overline{r^E})= \{P_1,P_3\}$),
- There exists an equivalence class represented by $\phi$. It includes queries that belong to no authorization space.
- We note that an element of $\wp(E)$ may not correspond to an authorization class. For example $\{P_1,P_4\}$, $\{P_2,P_4\}$, $\{P_1, P_3,P_4\}$ et $\{P_1,P_2,P_3,P_4\}$.
- One can deduce from the remarks above that there is a subset B of $\wp(E)$ such that for each element $r \in R_\Sigma$, there exists $b \in B$ and $P(\overline{r^E}) = b$.
- Authorization classes $\overline{r^E}$ with $card(P(\overline{r^E}))$ equals to 1 or 0 are not included in any conflict/anomaly.

In the following we study the impact of changing the initially provided set of policies on the canonical representation of the query space.

## 4.4 Impact of the modification of initialy provided policy set on the canonical representation

### 4.4.1 Policy addition

When adding a new policy to the policy repository, the canonical representation must be updated. Here we give some necessary results to this end:
***Proposition 3*** Let $E = \{P_1,P_2, \ldots, P_n\}$ and $E' = E \cup \{P_{n+1}\}$. Then $\forall r \in R_\Sigma$ :

- $\overline{r^E} = \overline{r^{E\prime}} \cup \overline{r_1^{E\prime}}$ with $r_1 \in (\overline{r^E} \setminus \overline{r^{E\prime}})$ if $(\overline{r^E} \setminus \overline{r^{E\prime}}) \neq \phi$
- or $\overline{r^E} = \overline{r^{E\prime}}$ otherwise.

In other words, an element of the canonical representation of $R_\Sigma$ in E is equal either:

- to the union of two elements of the canonical representation of $R_\Sigma$ in $E \cup \{P_{n+1}\}$ or,
- to an element of the canonical representation of $R_\Sigma$ in $E \cup \{P_{n+1}\}$,

***Proof***
Let $P(\overline{r^E}) = \{P'_1, \ldots ,P'_m\}$. By definition $\forall r \in \overline{r^E}$ $\{P'_1, \ldots ,P'_m\}$ are the only applicable policies to r in E.
To determine authorization classes in E' we must study the applicability of $P_{n+1}$ to requests in $\overline{r^E}$. Indeed, we have three cases depending on whether the policy $P_{n+1}$ is applicable or not to requests in the authorization class $\overline{r^E}$:
Case 1: $P_{n+1}$ is applicable to all requests in $\overline{r^E}$
Let $r' \in \overline{r^E}$. Then $\{P'_1, \ldots ,P'_m, P_{n+1}\}$ are applicable to r'. To prove that $r' \in \overline{r^{E\prime}}$ it suffices to prove that $\{P'_1, \ldots ,P'_m, P_{n+1}\}$ are the only applicable policies to r' in E'.
Suppose that there exists $P_j$ in $E \cup \{P_{n+1}\} \setminus \{P'_1, \ldots ,P'_m, P_{n+1}\}$ such that $P_j$ is applicable to r '. Then there exists $P_j$ in $E \setminus \{P'_1, \ldots ,P'_m\}$ such that $P_j$ is applicable to r'. This contradicts the assumption $r' \in \overline{r^E}$. Therefore $r' \in \overline{r^{E\prime}}$.
Hence $\overline{r^E} = \overline{r^{E\prime}}$.
Case 2 : $P_{n+1}$ is not applicable to any queries in $\overline{r^E}$
In the same way as case 1 we can show that $\overline{r^E} = \overline{r^{E\prime}}$.
Case 3: $P_{n+1}$ is applicable to a strict subset of $\overline{r^E}$
Let $r_1, r_2 \in \overline{r^E}$ such that $P_{n+1}$ is applicable to $r_1$ and not applicable to $r_2$. Then $P(\overline{r_2^{E\prime}}) = \{P'_1, \ldots ,P'_m\}$ and $P(\overline{r_1^{E\prime}}) = \{P'_1, \ldots ,P'_m, P_{n+1}\}$. It is clear that $\overline{r_1^{E\prime}} \cup \overline{r_2^{E\prime}} \subseteq \overline{r^E}$.

Inversely, let r' $\in \overline{r^E}$. Then {P'$_1$, … ,P'$_m$} are applicable to r'. If $P_{n+1}$ is applicable to r' then r' $\in \overline{r_1^{E\prime}}$ otherwise r' $\in \overline{r_2^{E\prime}}$. Therefore $\overline{r^E} \subseteq \overline{r_1^{E\prime}} \cup \overline{r_2^{E\prime}}$.

***Proposition 4*** let E = {$P_1$,$P_2$, …, $P_n$}, $P_i$ and $P_j$ are two policies of E, $f_i$ (resp. $f_j$) is the atomic Boolean expression of $P_i$ (resp. of $P_j$) on the attribute name *a*. Then, if for each value v in the domain of *a*, $f_i(v) \wedge f_j(v)$ = false then $P_1$ and $P_2$ can not belong to the same policy set defining an element of the canonical representation.

The proof is obvious.

### 4.4.2 Deleting a policy

The canonical representation must also be updated after the deletion of a policy from the policy repository.

Let E = {$P_1$,$P_2$, …, $P_n$} and E' = E $\cup$ $P_{n+1}$ two policy sets. Let CR (resp. CR') the canonical representation of $R_\Sigma$ with respect to E (resp. E'). One can deduce from Proposition 3 above that to reconstruct CR from CR', it suffices to combine each two elements of CR that have respective applicable policies of the form {P'$_1$, … ,P'$_m$, $P_{n+1}$} and {P'$_1$, … ,P'$_m$}.

Note that we did not deal here with the policy modification because it is equivalent to deleting a policy and then replacing it by a new one.

### 4.4.3 Calculation of the policy integration expression on canonical representation elements

In the above we have segmented the query space into subsets (authorization classes). Each class can be seen as the target of a policy (in a future conflict-free system). It remains to calculate the decision of this policy by combining different active policies on each authorization class.

Policy integration expressions introduced by the FIA algebra [23] provide a mechanism richer than the standard GeoXACML combining algorithms. In the next proposition we calculate the projection of the FIA policy integration expression on the different authorization classes (that we call *the* resultant policy integration expression). The final goal is to bring the study of conflict resolution from $R_\Sigma$ to its canonical representation.

***Proposition 5*** Let E = {$P_1$,$P_2$, …, $P_n$} be a set of policies and dc a domain constraint. Let ($\prod_{dc}$(E) be the set of policy projections on dc defined as follows : $\prod_{dc}(E) = \{\prod_{dc}(P) | P \in E\}$. let f be a FIA policy integration expression in E. Then there exists a policy integration expression $f_{dc}$ in $\prod_{dc}(E)$ such that $\prod_{dc}(f(P_1,P_2, \ldots, P_n)) = f_{dc}(\prod_{dc}(P_1), \prod_{dc}(P_2), \ldots, \prod_{dc}(P_n))$

***Proof***

According to the "minimum set of operators" theorem (see [23]), it suffices to prove the proposition for the following operators $\{\neg, P_Y, \&, +, \Pi_{dC}\}$.

Suppose that f consists of a single monomial then f can be written in one of the following formats:

$f(P_1,P_2, \ldots, P_n) = P_i$,

$f(P_1,P_2, \ldots, P_n) = \neg P_i$,

$f(P_1,P_2, \ldots, P_n) = \prod_{dc1}(P_i)$,

$f(P_1,P_2, \ldots, P_n) = P_Y$

Then $f_{dc}$ will be defined respectively as follows:

$f_{dc}(\prod_{dc}(P_1), \prod_{dc}(P_2), \ldots, \prod_{dc}(P_n)) = \prod_{dc}(P_i)$,

$f(\prod_{dc}(P_1), \prod_{dc}(P_2), \ldots, \prod_{dc}(P_n)) = \neg\prod_{dc}(P_i)$,

$f(\prod_{dc}(P_1), \prod_{dc}(P_2), \ldots, \prod_{dc}(P_n)) = \prod_{dc1}(\prod_{dc}(P_i))$,

$f(\prod_{dc}(P_1), \prod_{dc}(P_2), \ldots, \prod_{dc}(P_n)) = P_Y$.

It is clear that in all cases:

$\prod_{dc}(f(P_1, P2, \ldots, P_n)) = f_{dc}(\prod_{dc}(P_1), \prod_{dc}(P_2), \ldots, \prod_{dc}(P_n))$.

Suppose now that the result is true for an expression of m monomials and show that this remains true for an expression of a (m+1) monomials:

Let f be an expression of (m+1) monomials. Then there exists an expression f' of m monomials such that:

$f(P_1,P_2, \ldots, P_m, P_{m+1}) = f'(P_1,P_2, \ldots, P_m)$ <binary_opetrator> (<unary_ opetrator > $P_{m+1}$)

We define $f_{dc}$ as follows $f_{dc} = f'_{dc}$ <binary_opetrator> (<unary_ opetrator > $\prod_{dc}(P_{m+1})$)

$\prod_{dc}(f(P_1,P_2, \ldots, P_m, P_{m+1})) = \prod_{dc}(f'(P_1,P_2, \ldots, P_m)$ <binary_opetrator > (<unary_ opetrator > $P_{m+1}$))

$\prod_{dc}(f(P_1,P_2, \ldots, P_m, P_{m+1})) = \prod_{dc}(f'(P_1,P_2, \ldots, P_m))$ <binary_opetrator > $\prod_{dc}$ (<unary_ opetrator > $P_{m+1}$)

According to the recurrence hypothesis : $\prod_{dc}(f'(P_1,P_2, \ldots, P_m)) = f'_{dc}(\prod_{dc}(P_1), \prod_{dc}(P_2), \ldots, \prod_{dc}(P_m))$

$\prod_{dc}(f(P_1,P_2, \ldots, P_m, P_{m+1}))= f'_{dc} (\prod_{dc}(P_1), \prod_{dc}(P_2), \ldots , \prod_{dc}(P_m))$ < binary_opetrator > (<unary_ opetrator> $\prod_{dc}(P_{m+1})$)

$\prod_{dc}(f(P_1,P_2, \ldots, P_m, P_{m+1}))= f_{dc} (\prod_{dc}(P_1), \prod_{dc}(P_2), \ldots , \prod_{dc}(P_m) , \prod_{dc}(P_{m+1}))$.

To evaluate the resultant policy integration expression in an element of the canonical representation, we propose an algorithm based on binary expression trees. It will comprise two steps:

- Step 1: This step can be performed by one of the classical algorithms of construction of binary expression trees from an arithmetic expression. So it will have as input the FIA policy integration expression and the result will be the binary expression tree.
- Step 2: the input of this step is the binary tree constructed in Step 1. The algorithm substitutes policies by their projections in the binary tree and then reduces this tree by exploiting the arithmetic properties of the policy $P_{NA}$ (such as: $\forall$ P, $P_{NA}$ & P = $P_{NA}$, P + $P_{NA}$ =P … ). The output of this step is the resultant policy integration expression.

## 5. CONCLUSION

Securing geospatial data is a crucial issue in a wide range of applications. In this context, our contribution examines the problem of conflicts between GEOXACML policies controlling access to geographic information systems. We gave a special attention to the mathematical formalism of the problem. The canonical representation of the query space proposed in this paper can be used to construct a conflict-free set of policies derived from the initial policy set. This component can form the core of a Framework for detection and resolution of conflicts between GEOXACML policies. This will be the subject of a future work.

## REFERENCES


[1] Abedin, M. et al, 2006. Detection and resolution of anomalies in firewall policy rules. *Proceedings 20th Annual IFIP WG 11.3, Working Conference on Data and Applications Security (DBSec)*. pp. 15-29.

[2] Anderson, A., 2003. Evaluating xacml as a policy language. *Technical report*. OASIS.

[3] Ardagna, C. A. et al, 2006. A Web Service Architecture for Enforcing Access Control Policies. *Electronic Notes in Theoretical Computer Science*, Vol. 142, pp 47-62.

[4] Bertino, E. and Damiani, M. L. 2004. A Controlled Access to Spatial Data on Web. *Proceedings of the 14th 7th AGILE Conference on Geographic Information Science*. Heraklion, Greece, pp. 82-91.

[5] Bertino, E. et al, 2003. A Logical Framework for Reasoning about Access Control Models. *ACM Transactions on Information and System Security*, Vol. 6, No. 1, pp.71–127.

[6] Bertino, E. et al, 2008. Security and Privacy for Geospatial Data: Concepts and Research Directions, *Proceedings of the SIGSPATIAL ACM GIS: International Workshop on Security and Privacy in GIS and LBS*. Irvine, USA, pp. 6-19

[7] Bertino, E., et al, 2004. An authorization model for geographical maps. *Proceedings of the 12th annual ACM international workshop on Geographic information systems*. Washington, DC, USA. pp. 82-91.

[8] Bonatti, P. et al, 2002. An algebra for composing access control policies. *ACM Transactions on Information and System Security (TISS)*, Vol. 5, No. 1, pp. 1–35.

[9] Bryant, R.1986. Graph-based algorithms for boolean function manipulation. *IEEE Transactions on computers*, C-35,8, pp. 677–691.

[10] Fisler, K. et al, 2005. Verification and change-impact analysis of access-control policies. *Proceedings of the 27th international conference on Software engineering*, pp. 196–205.

[11] Halpern, J. Y. and Weissman, V. 2008. Using first-order logic to reason about policies. *Computer Security Foundations Workshop in Pacific Grove*. California, USA. pp 187–201

[12] Herrmann, J. 2010. Access control systems for spatial data infrastructures and their administration. *Proceedings of Proceedings of the 4th ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS.* Chicago, USA, pp. 53-59.

[13] Hu, H. et al, 2011. Anomaly Discovery and Resolution in Web Access Control Policies. *Proceedings of the 16th ACM Symposium on Access Control Models and Technologies: SACMAT 11*. Innsbruck, Austria, pp.165-174.

[14] Hu, H. et al, 2011. Ontology-based Policy Anomaly Management for Autonomic Computing. *Proceedings of 7th International Conference on Collaborative Computing*. Orlando, Florida, USA, pp.487-494.

[15] Hughes, G. and Bultan, T., 2004. Automated verification of access control policies. *Technical Report 2004-22, Department of Computer Science, University of California,* Santa Barbara,US.

[16] Kolovski, V. 2008. *A Logic-Based Framework for Web Access Control Policies*. PhD thesis, University of Maryland.

[17] Landwehr, C. 1981. Formal Models for Computer Security. *ACM Computing Surveys (CSUR)*, pp. 247-278.

[18] Matheus, A. 2005. Declaration and enforcement of fine-grained access restrictions for a service-based geospatial data infrastructure. *Proceedings of the 10th ACM symposium on Access control models and technologies*. Stockholm, Sweden. pp. 21-28.

[19] Matheus, A. and Herrmann, J. 2008. *Geospatial eXtensible Access Control Markup Language (GeoXACML)*. OGC Implementation Standard, Open Geospatial Consortium.

[20] Nebert, D. 2004. Developing Spatial Data Infrastructures: The SDI Cookbook. GSDI.

[21] Ni, Q. et al, 2009. D-algebra for composing access control policy decisions. *Proceedings of ACM Symposium on Information, Computer and Communication Symposium (ASIACCS 2009)*. Sydney, Australia, pp. 298-309.

[22] OASIS. 2005. Extensible access control markup language (xacml). *OASIS Standard*, version 2.0.

[23] Rao, P. et al, 2009. An algebra for fine-grained integration of XACML policies. *Proceedings of the 14th ACM symposium on Access control models and technologies : SACMAT 09*. Streasa, Italy, pp.63-72.

[24] Thion, R. 2008. Structuration relationnelle des politiques de contrôle d'accès : Représentation, raisonnement et vérification logiques. Thèse de doctorat, Institut national des sciences appliquées de Lyon. pp 67-47

[25] Yahiaoui, M. et al, 2012. Formal representation of conflict zones in XACML access control systems. *Colloquium on Information Science and Technology*. Fez, Morooco, pp. 123-129.

[26] Yahiaoui, M. et al, 2013. A comprehensive framework for managing conflicts/anomalies between XACML policies: mathematical basis and architecture. *Journal of Theoretical and Applied Information Technology*, Vol. 49, No.3, pp. 955-969.