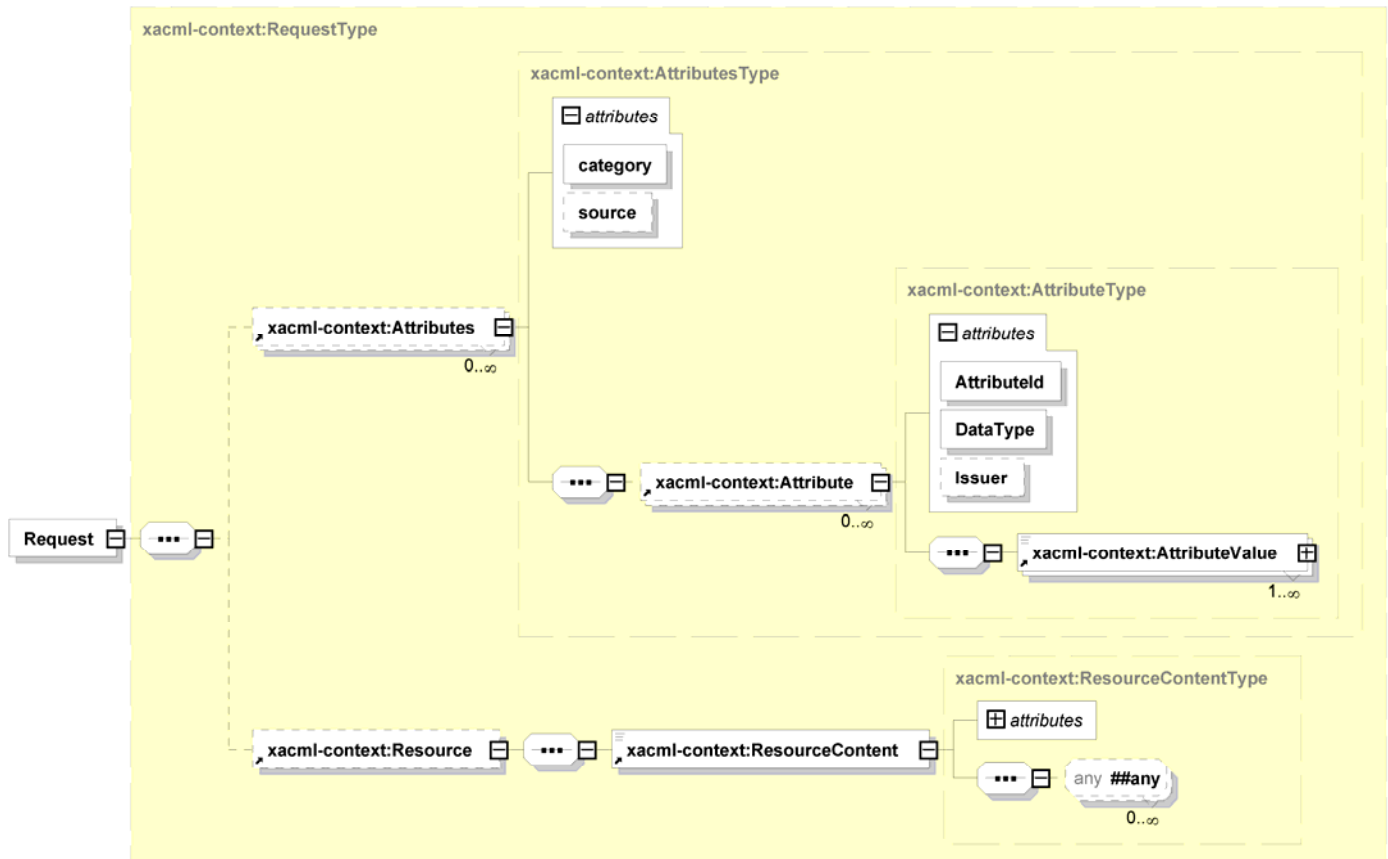


Introducing attribute categories.

1. The purpose of this proposal is to add extensible attribute categories to XACML context.
2. Current specification implicitly defines following attribute categories
 - 2.1. Multiple subject categories differentiated by Subject Category URI.
 - 2.1.1. Default subject category is :
`urn:oasis:names:tc:xacml:1.0:subject-category:access-subject`
 - 2.2. Resource category. Multiple resource groups may be specified in request
 - 2.3. Action category
 - 2.4. Environment category
3. In this proposal all existing attribute categories in request and policy are substituted with a uniform specification.
 - 3.1. New reserved URIs are introduced to represent existing categories
 - 3.1.1. `urn:oasis:names:tc:xacml:3.0:context-category:resource`
 - 3.1.2. `urn:oasis:names:tc:xacml:3.0:context-category:action`
 - 3.1.3. `urn:oasis:names:tc:xacml:3.0:context-category:environment`
 - 3.1.4. Existing subject categories are reused, including
`urn:oasis:names:tc:xacml:1.0:subject-category:access-subject`

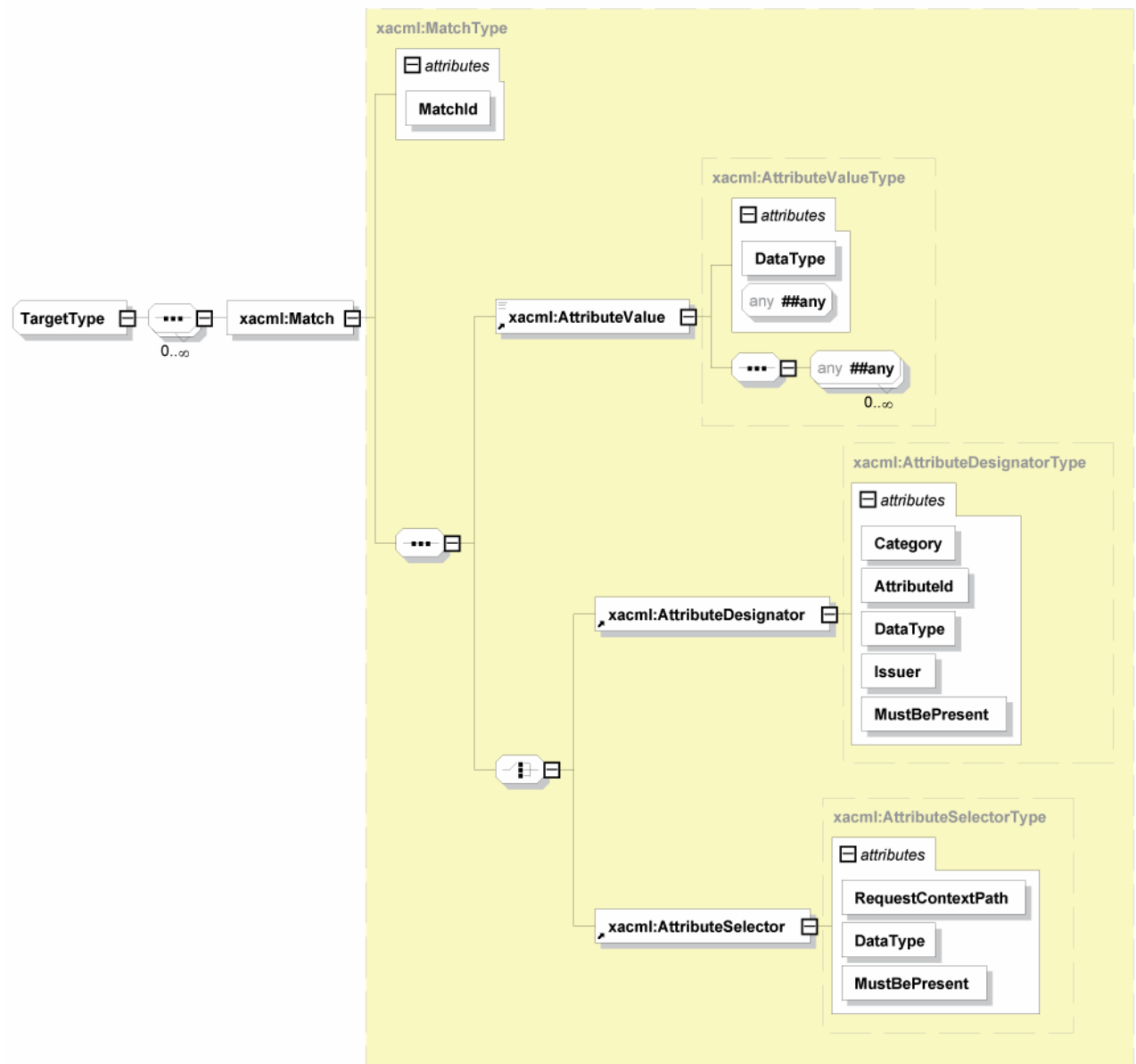
Note: Shorter names, such as `xacml-context:action` may be more user friendly.
 - 3.2. XACML 2.0 requests and policies may always be translated into the new representation without any loss of functionality
 - 3.2.1. XPath expressions used to identify values in request context other than under Request/Resource/ResourceContent path will need to be changed to use a predicated path step expression instead.
 - 3.2.1.1. For example
`/Request/Action/Attribute`
will become (and here is where a shorter URI may be friendly)
`/Request/Attributes[@category =
"urn:oasis:names:tc:xacml:3.0:context-category:action"]/Attribute`
 - 3.3. Definitions of resource attribute, subject attribute need to be changed to refer to an attribute in a specific category, rather than in a particular element.
4. Request schema:
 - 4.1. `<Resource>` and `<ResourceContent>` elements are preserved, so that path expressions used to identify parts of the resource content in XACML 2.0 need not to be changed.

- 4.2. Hierarchical and multiple resource profiles will need to be modified to change the definition of resource attribute and remove references to the <Resource> element, and substitute them with references to the resource attribute category.



5. Target schema

- 5.1. AttributeDesignator is the only remaining element – all Subject/Resource/Action/Environment designator are removed. They are substituted with a reference to the attribute category in the AttributeDesignator.
- 5.1.1. To preserve some compatibility with 2.0 policies one may keep the <XxxAttributeDesignator> types and just make them derived by restriction from AttributeDesignator by fixing the value of the “Category” attribute to one of the predefined categories.



6. Changes to the document:

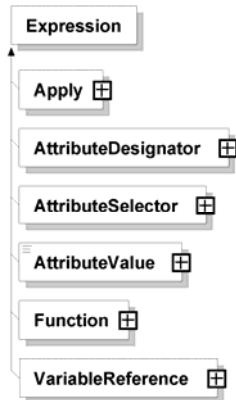
6.1. Remove sections 5.6 to 5.17 and introduce single <Match> element:

6.1.1. Element <Target> SHALL contain a disjunctive sequence of <Match> elements.

6.1.2. Element <Match> SHALL identify a target by matching attribute values in the <Attributes> element of the request context with a specified category with the embedded attribute value.

6.2. Remove sections 6.1 to 6.6 and add a description of <Attributes> element

- 6.2.1. The <Attributes> element specifies attributes in a specified category. For example `urn:oasis:names:tc:xacml:3.0:context-category:action` category specifies attributes associated with the action by listing a set of <Attribute> elements.
- 6.3. Expression substitution group in section 7.3 should be modified to use only one attribute designator.



- 6.4. Sections 5.38 to 5.41 should be removed and section 5.37 rewritten

Element <AttributeDesignator> is used to identify attributes by name and category. It contains information to match attributes in the request context. It also contains information in the event that no matching attribute is present in the context.

A named attribute SHALL match an attribute if the value of their respective Category matches the Category attribute of the <Attributes> element by URI equality, and AttributeId, DataType and Issuer match their respective values for the context attribute.

If the Issuer attribute is present in the attribute designator, then it MUST match, using the “urn:oasis:names:tc:xacml:1.0:function:string-equal” function, the Issuer of the same attribute. If the Issuer is not present in the attribute designator, then the matching of the attribute to the named attribute SHALL be governed by AttributeId and DataType attributes alone.

The <AttributeDesignator> element retrieves a bag of values for a named categorized attribute from the request context. A categorized attribute is an attribute contained within any of the <Attributes> elements in the request context with the matching Category attribute. If more than one <Attributes> elements with the same category are present they shall be treated as one element.

The <AttributeDesignator> element SHALL return a bag containing all the attribute values that are matched by the named categorized attribute. In the event that no matching attribute is present in the context, the MustBePresent attribute governs whether this element returns an empty bag or “Indeterminate”.

The <AttributeDesignator> contains the following attributes:

Category [Required]

This attribute SHALL specify the Category in which to match attribute

AttributeId [Required]

This attribute SHALL specify the AttributeId with which to match the attribute.

DataType [Required]

The bag returned by the <AttributeDesignator> element SHALL contain values of this data-type.

Issuer [Optional]

This attribute, if supplied, SHALL specify the Issuer with which to match the attribute.

MustBePresent [Optional]

This attribute governs whether the element returns “Indeterminate” or an empty bag in the event the named attribute is absent from the request context.

Context schema

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2006 sp2 U (http://www.altova.com) by Daniel
Engovator (W3C) -->
<xs:schema xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
xmlns:xacml-context="urn:oasis:names:tc:xacml:2.0:context:schema:os"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:oasis:names:tc:xacml:2.0:context:schema:os"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
schemaLocation="http://docs.oasis-open.org/xacml/access_control-xacml-2.0-policy-
schema-os.xsd"/>
  <!-- -->
  <xs:element name="Request" type="xacml-context:RequestType"/>
  <xs:complexType name="RequestType">
    <xs:sequence>
      <xs:element ref="xacml-context:Attributes" minOccurs="0"
maxOccurs="unbounded"/>
      <xs:element ref="xacml-context:Resource" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <!-- -->
  <xs:element name="Response" type="xacml-context:ResponseType"/>
  <xs:complexType name="ResponseType">
    <xs:sequence>
      <xs:element ref="xacml-context:Result" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <!-- -->
  <xs:element name="Resource">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="xacml-context:ResourceContent"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="ResourceContent" type="xacml-context:ResourceContentType"/>
  <xs:complexType name="ResourceContentType" mixed="true">
    <xs:sequence>
      <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```

        </xs:sequence>
        <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:complexType>
    <!-- -->
    <xs:element name="Attributes" type="xacml-context:AttributesType"/>
    <xs:complexType name="AttributesType">
        <xs:sequence>
            <xs:element ref="xacml-context:Attribute" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="category" type="xs:anyURI" use="required"/>
        <xs:attribute name="source" type="xs:anyURI" use="optional"/>
    </xs:complexType>
    <!-- -->
    <xs:element name="Attribute" type="xacml-context:AttributeType"/>
    <xs:complexType name="AttributeType">
        <xs:sequence>
            <xs:element ref="xacml-context:AttributeValue"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="AttributeId" type="xs:anyURI" use="required"/>
        <xs:attribute name="DataType" type="xs:anyURI" use="required"/>
        <xs:attribute name="Issuer" type="xs:string" use="optional"/>
    </xs:complexType>
    <!-- -->
    <xs:element name="AttributeValue" type="xacml-context:AttributeValueType"/>
    <xs:complexType name="AttributeValueType" mixed="true">
        <xs:sequence>
            <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:anyAttribute namespace="##any" processContents="lax"/>
    </xs:complexType>
    <!-- -->
    <xs:element name="Result" type="xacml-context:ResultType"/>
    <xs:complexType name="ResultType">
        <xs:sequence>
            <xs:element ref="xacml-context:Decision"/>
            <xs:element ref="xacml-context:Status" minOccurs="0"/>
            <xs:element ref="xacml:Obligations" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="ResourceId" type="xs:string" use="optional"/>
    </xs:complexType>
    <!-- -->
    <xs:element name="Decision" type="xacml-context:DecisionType"/>
    <xs:simpleType name="DecisionType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="Permit"/>
            <xs:enumeration value="Deny"/>
            <xs:enumeration value="Indeterminate"/>
            <xs:enumeration value="NotApplicable"/>
        </xs:restriction>
    </xs:simpleType>
    <!-- -->
    <xs:element name="Status" type="xacml-context:StatusType"/>
    <xs:complexType name="StatusType">
        <xs:sequence>
            <xs:element ref="xacml-context:StatusCode"/>
            <xs:element ref="xacml-context:StatusMessage" minOccurs="0"/>
            <xs:element ref="xacml-context:StatusDetail" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <!-- -->
    <xs:element name="StatusCode" type="xacml-context:StatusCodeType"/>
    <xs:complexType name="StatusCodeType">
        <xs:sequence>
            <xs:element ref="xacml-context:StatusCode" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="Value" type="xs:anyURI" use="required"/>
    </xs:complexType>
    <!-- -->
    <xs:element name="StatusMessage" type="xs:string"/>
    <!-- -->
    <xs:element name="StatusDetail" type="xacml-context:StatusDetailType"/>
    <xs:complexType name="StatusDetailType">
        <xs:sequence>
            <xs:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <!-- -->

```

```

        <xs:element name="MissingAttributeDetail" type="xacml-
context:MissingAttributeDetailType"/>
        <xs:complexType name="MissingAttributeDetailType">
            <xs:sequence>
                <xs:element ref="xacml-context:AttributeValue" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="AttributeId" type="xs:anyURI" use="required"/>
            <xs:attribute name="DataType" type="xs:anyURI" use="required"/>
            <xs:attribute name="Issuer" type="xs:string" use="optional"/>
        </xs:complexType>
    <!-- -->
</xs:schema>

```

Policy schema

(Current schema does not validate in several tools)

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2006 sp2 U (http://www.altova.com) by Daniel
Engovатов (W3C) -->
<xs:schema xmlns:xacml="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <!-- -->
    <xs:element name="PolicySet" type="xacml:PolicySetType"/>
    <xs:complexType name="PolicySetType">
        <xs:sequence>
            <xs:element ref="xacml:Description" minOccurs="0"/>
            <xs:element ref="xacml:PolicySetDefaults" minOccurs="0"/>
            <xs:element ref="xacml:Target"/>
            <xs:choice minOccurs="0" maxOccurs="unbounded">
                <xs:element ref="xacml:PolicySet"/>
                <xs:element ref="xacml:Policy"/>
                <xs:element ref="xacml:PolicySetIdReference"/>
                <xs:element ref="xacml:PolicyIdReference"/>
                <xs:element ref="xacml:CombinerParameters"/>
                <xs:element ref="xacml:PolicyCombinerParameters"/>
            </xs:choice>
            <xs:element ref="xacml:PolicySetCombinerParameters"/>
            <xs:element ref="xacml:Obligations" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="PolicySetId" type="xs:anyURI" use="required"/>
        <xs:attribute name="Version" type="xacml:VersionType"
default="1.0"/>
        <xs:attribute name="PolicyCombiningAlgId" type="xs:anyURI"
use="required"/>
    </xs:complexType>
    <!-- -->
    <xs:element name="CombinerParameters"
type="xacml:CombinerParametersType"/>
    <xs:complexType name="CombinerParametersType">
        <xs:sequence>
            <xs:element ref="xacml:CombinerParameter" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <!-- -->
    <xs:element name="CombinerParameter" type="xacml:CombinerParameterType"/>
    <xs:complexType name="CombinerParameterType">
        <xs:sequence>
            <xs:element ref="xacml:AttributeValue"/>
        </xs:sequence>
        <xs:attribute name="ParameterName" type="xs:string"
use="required"/>
    </xs:complexType>
    <!-- -->
    <xs:element name="RuleCombinerParameters"
type="xacml:RuleCombinerParametersType"/>
    <xs:complexType name="RuleCombinerParametersType">
        <xs:complexContent>
            <xs:extension base="xacml:CombinerParametersType">

```

```

                                <xs:attribute name="RuleIdRef" type="xs:string"
use="required"/>
                                </xs:extension>
                                </xs:complexContent>
                                </xs:complexType>
                                <!-- -->
                                <xs:element name="PolicyCombinerParameters"
type="xacml:PolicyCombinerParametersType"/>
                                <xs:complexType name="PolicyCombinerParametersType">
                                    <xs:complexContent>
                                        <xs:extension base="xacml:CombinerParametersType">
                                            <xs:attribute name="PolicyIdRef" type="xs:anyURI"
use="required"/>
                                        </xs:extension>
                                    </xs:complexContent>
                                </xs:complexType>
                                <!-- -->
                                <xs:element name="PolicySetCombinerParameters"
type="xacml:PolicySetCombinerParametersType"/>
                                <xs:complexType name="PolicySetCombinerParametersType">
                                    <xs:complexContent>
                                        <xs:extension base="xacml:CombinerParametersType">
                                            <xs:attribute name="PolicySetIdRef" type="xs:anyURI"
use="required"/>
                                        </xs:extension>
                                    </xs:complexContent>
                                </xs:complexType>
                                <!-- -->
                                <xs:element name="PolicySetIdReference" type="xacml:IdReferenceType"/>
                                <xs:element name="PolicyIdReference" type="xacml:IdReferenceType"/>
                                <!-- -->
                                <xs:element name="PolicySetDefaults" type="xacml:DefaultsType"/>
                                <xs:element name="PolicyDefaults" type="xacml:DefaultsType"/>
                                <xs:complexType name="DefaultsType">
                                    <xs:sequence>
                                        <xs:choice>
                                            <xs:element ref="xacml:XPathVersion"/>
                                        </xs:choice>
                                    </xs:sequence>
                                </xs:complexType>
                                <!-- -->
                                <xs:element name="XPathVersion" type="xs:anyURI"/>
                                <!-- -->
                                <xs:complexType name="IdReferenceType">
                                    <xs:simpleContent>
                                        <xs:extension base="xs:anyURI">
                                            <xs:attribute name="Version"
type="xacml:VersionMatchType" use="optional"/>
                                            <xs:attribute name="EarliestVersion"
type="xacml:VersionMatchType" use="optional"/>
                                            <xs:attribute name="LatestVersion"
type="xacml:VersionMatchType" use="optional"/>
                                        </xs:extension>
                                    </xs:simpleContent>
                                </xs:complexType>
                                <!-- -->
                                <xs:simpleType name="VersionType">
                                    <xs:restriction base="xs:string">
                                        <xs:pattern value="(\d+\.)*\d+"/>
                                    </xs:restriction>
                                </xs:simpleType>
                                <!-- -->
                                <xs:simpleType name="VersionMatchType">
                                    <xs:restriction base="xs:string">
                                        <xs:pattern value="((\d+|\*)\.)*(\d+|\*|\+)/"/>
                                    </xs:restriction>
                                </xs:simpleType>
                                <!-- -->
                                <xs:element name="Policy" type="xacml:PolicyType"/>
                                <xs:complexType name="PolicyType">
                                    <xs:sequence>
                                        <xs:element ref="xacml:Description" minOccurs="0"/>
                                        <xs:element ref="xacml:PolicyDefaults" minOccurs="0"/>
                                        <xs:element ref="xacml:CombinerParameters" minOccurs="0"/>
                                        <xs:element ref="xacml:Target"/>
                                        <xs:choice maxOccurs="unbounded">
                                            <xs:element ref="xacml:CombinerParameters"
minOccurs="0"/>
                                            <xs:element ref="xacml:RuleCombinerParameters"
minOccurs="0"/>
                                            <xs:element ref="xacml:VariableDefinition"/>

```

```

        <xs:element ref="xacml:Rule" />
    </xs:choice>
    <xs:element ref="xacml:Obligations" minOccurs="0" />
</xs:sequence>
<xs:attribute name="PolicyId" type="xs:anyURI" use="required" />
<xs:attribute name="Version" type="xacml:VersionType"
default="1.0" />
    <xs:attribute name="RuleCombiningAlgId" type="xs:anyURI"
use="required" />
</xs:complexType>
<!-- -->
<xs:element name="Description" type="xs:string" />
<!-- -->
<xs:element name="Rule" type="xacml:RuleType" />
<xs:complexType name="RuleType">
    <xs:sequence>
        <xs:element ref="xacml:Description" minOccurs="0" />
        <xs:element ref="xacml:Target" minOccurs="0" />
        <xs:element ref="xacml:Condition" minOccurs="0" />
    </xs:sequence>
    <xs:attribute name="RuleId" type="xs:string" use="required" />
    <xs:attribute name="Effect" type="xacml:EffectType"
use="required" />
</xs:complexType>
<!-- -->
<xs:simpleType name="EffectType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Permit" />
        <xs:enumeration value="Deny" />
    </xs:restriction>
</xs:simpleType>
<!-- -->
<xs:element name="Target" type="xacml:TargetType" />
<xs:complexType name="TargetType">
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:element name="Match" type="xacml:MatchType" />
    </xs:sequence>
</xs:complexType>
<!-- -->
<xs:element name="Match" type="xacml:MatchType" />
<xs:complexType name="MatchType">
    <xs:sequence>
        <xs:element ref="xacml:AttributeValue" />
        <xs:choice>
            <xs:element ref="xacml:AttributeDesignator" />
            <xs:element ref="xacml:AttributeSelector" />
        </xs:choice>
    </xs:sequence>
    <xs:attribute name="MatchId" type="xs:anyURI" use="required" />
</xs:complexType>
<!-- -->
<xs:element name="VariableDefinition"
type="xacml:VariableDefinitionType" />
<xs:complexType name="VariableDefinitionType">
    <xs:sequence>
        <xs:element ref="xacml:Expression" />
    </xs:sequence>
    <xs:attribute name="VariableId" type="xs:string" use="required" />
</xs:complexType>
<!-- -->
<xs:element name="Expression" abstract="true">
    <xs:complexType>
        <xs:complexContent>
            <xs:extension base="xacml:ExpressionType" />
        </xs:complexContent>
    </xs:complexType>
</xs:element>
<xs:complexType name="ExpressionType" abstract="true" />
<!-- -->
<xs:element name="VariableReference" type="xacml:VariableReferenceType"
substitutionGroup="xacml:Expression" />
<xs:complexType name="VariableReferenceType">
    <xs:complexContent>
        <xs:extension base="xacml:ExpressionType">
            <xs:attribute name="VariableId" type="xs:string"
use="required" />
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
<!-- -->

```

```

        <xs:element name="AttributeSelector" type="xacml:AttributeSelectorType"
substitutionGroup="xacml:Expression"/>
        <xs:complexType name="AttributeSelectorType">
            <xs:complexContent>
                <xs:extension base="xacml:ExpressionType">
                    <xs:attribute name="RequestContextPath"
type="xs:string" use="required"/>
                    <xs:attribute name="DataType" type="xs:anyURI"
use="required"/>
                    <xs:attribute name="MustBePresent" type="xs:boolean"
use="optional" default="false"/>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
        <!-- -->
        <xs:element name="AttributeDesignator"
type="xacml:AttributeDesignatorType" substitutionGroup="xacml:Expression"/>
        <xs:complexType name="AttributeDesignatorType">
            <xs:complexContent>
                <xs:extension base="xacml:ExpressionType">
                    <xs:attribute name="Category" type="xs:anyURI"
use="required"/>
                    <xs:attribute name="AttributeId" type="xs:anyURI"
use="required"/>
                    <xs:attribute name="DataType" type="xs:anyURI"
use="required"/>
                    <xs:attribute name="Issuer" type="xs:string"
use="optional"/>
                    <xs:attribute name="MustBePresent" type="xs:boolean"
use="optional" default="false"/>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
        <!-- -->
        <xs:element name="AttributeValue" type="xacml:AttributeValueType"
substitutionGroup="xacml:Expression"/>
        <xs:complexType name="AttributeValueType" mixed="true">
            <xs:complexContent mixed="true">
                <xs:extension base="xacml:ExpressionType">
                    <xs:sequence>
                        <xs:any namespace="##any"
processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
                    </xs:sequence>
                    <xs:attribute name="DataType" type="xs:anyURI"
use="required"/>
                    <xs:anyAttribute namespace="##any"
processContents="lax"/>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
        <!-- -->
        <xs:element name="Function" type="xacml:FunctionType"
substitutionGroup="xacml:Expression"/>
        <xs:complexType name="FunctionType">
            <xs:complexContent>
                <xs:extension base="xacml:ExpressionType">
                    <xs:attribute name="FunctionId" type="xs:anyURI"
use="required"/>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
        <!-- -->
        <xs:element name="Condition" type="xacml:ConditionType"/>
        <xs:complexType name="ConditionType">
            <xs:sequence>
                <xs:element ref="xacml:Expression"/>
            </xs:sequence>
        </xs:complexType>
        <!-- -->
        <xs:element name="Apply" type="xacml:ApplyType"
substitutionGroup="xacml:Expression"/>
        <xs:complexType name="ApplyType">
            <xs:complexContent>
                <xs:extension base="xacml:ExpressionType">
                    <xs:sequence>
                        <xs:element ref="xacml:Expression"
minOccurs="0" maxOccurs="unbounded"/>
                    </xs:sequence>
                    <xs:attribute name="FunctionId" type="xs:anyURI"
use="required"/>
                </xs:extension>
            </xs:complexContent>
        </xs:complexType>
    </xs:sequence>

```

```

        </xs:complexContent>
    </xs:complexType>
    <!-- -->
    <xs:element name="Obligations" type="xacml:ObligationsType"/>
    <xs:complexType name="ObligationsType">
        <xs:sequence>
            <xs:element ref="xacml:Obligation" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <!-- -->
    <xs:element name="Obligation" type="xacml:ObligationType"/>
    <xs:complexType name="ObligationType">
        <xs:sequence>
            <xs:element ref="xacml:AttributeAssignment" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="ObligationId" type="xs:anyURI" use="required"/>
        <xs:attribute name="FulfillOn" type="xacml:EffectType"
use="required"/>
    </xs:complexType>
    <!-- -->
    <xs:element name="AttributeAssignment"
type="xacml:AttributeAssignmentType"/>
    <xs:complexType name="AttributeAssignmentType" mixed="true">
        <xs:complexContent mixed="true">
            <xs:extension base="xacml:AttributeValueType">
                <xs:attribute name="AttributeId" type="xs:anyURI"
use="required"/>
            </xs:extension>
        </xs:complexContent>
    </xs:complexType>
    <!-- -->
</xs:schema>

```