

# Attribute-based access control policy specification language

Ye Chunxiao Zhong Jiang Feng Yong

(College of Computer Science, Chongqing University, Chongqing 400044, China)

**Abstract:** This paper first introduces attribute expression to describe attribute-based access control policy. Secondly, an access control policy enforcement language named A-XACML (attribute-XACML) is proposed, which is an extension of XACML. A-XACML is used as a simple, flexible way to express and enforce access control policies, especially attribute-based access control policy, in a variety of environments. The language and schema support include data types, functions, and combining logic which allow simple and complex policies to be defined. Finally, a system architecture and application case of user-role assignment is given to show how attribute expressions and A-XACML work in access control policy description and enforcement. The case shows that attribute expression and A-XACML can describe and enforce the complex access control policy in a simple and flexible way.

**Key words:** role-based access control; policy; XML; XACML

Role-based access control (RBAC) has gained considerable attention recently<sup>[1]</sup>. Bhatti et al.<sup>[2]</sup> proposed an XML-based RBAC policy specification framework named X-RBAC for enforcing access control in dynamic XML-based web services. XML is used as an access control language allowing specification of RBAC policies and facilitating specification of timing constraints on roles and access requirements<sup>[3]</sup>. XACML is an OASIS standard that defines an architecture, policies and messages within an access control system<sup>[4]</sup>. Toktar et al.<sup>[5]</sup> used the XACML to model and distribute RSVP access control policies for RSVP-aware application servers.

For a better access control security, we use a user's and a role's attribute expression to describe access control policy. A user's attribute expression indicates a user's qualifications and abilities, and a role's attribute expression indicates a user's qualifications and abilities required by the role in a user-role assignment. Unfortunately, none of these researches fits for attribute access control.

## 1 Attribute and Attribute Expression

We first give two definitions for attribute-based access control. In definition 1, “{ }” means repetition, “+” means repeat one or more times, and “|” means only one value can be selected.

### Definition 1

UAE (user attribute expression) : : = uae | uae { AND uae } +  
RAE (role attribute expression) : : = ae | ae { AND ae } +

ae (attribute expression) : : = ua oprt uav  
uae (user attribute expression) : : = ua roprt uav  
oprt (operator) : : = “<” | “≤” | “=” | “≥” | “>” | “≠”  
roprt (relation operator) : : = “≠” | “=”  
ua (user attribute) : : = {specified by system}  
uav (user attribute value) : : = {specified by system}

where AND is the usual logic operator “and”.

For example, level = 4, type = “S” and total ≥ 33 are uaes or aes. level = 4 AND type = “S” is a UAE or an RAE. If a uae and an ae have the same attribute name, then we say that the uae and the ae are comparable. Similar to a recent study<sup>[6]</sup>, we use “▷” to denote the dominance relationships between two comparable uae and ae. For example, age = 30 ▷ age > 20 for 30 > 20 and age = 30 ▷ age < 40 for 30 < 40. We also use “▷” to denote the dominance relationship between a UAE and an RAE.

**Definition 2** We say UAE ▷ RAE, if  $\forall ae_j \in U\_RAE, \exists uae_i \in U\_UAE, s. t. uae_i \triangleright ae_j$ . Where  $U\_UAE, U\_RAE$  are the uae sets of a UAE and the ae sets of an RAE, respectively.

For example, UAE (level = 5 AND age = 20) ▷ RAE (age > 15) for age = 20 ▷ age > 15.

## 2 A-XACML

Based on XACML, we propose a policy language named A-XACML, which uses BNF to define elements. Similar to BNF, we use < ! -- element name > to define a non-terminal element and < tag name > or < /tag name > to define a terminal element. In A-XACML, “( )” means the context included in it must be filled by the system administrator, “[ ]” means optional.

**Definition 3** We define ua, uav, roprt and oprt in A-XACML as follows:

< ! -- ua > : : = < Apply FunctionId = “& function; any-of” > < Function FunctionId = “& function; String-equal” > /> < AttributeValue DataType = “&XML; String” > ( ua ) < /AttributeValue > < SubjectAttributeDesignator AttributeId = “(attributeid)” DataType = “&XML; String” > /> < /Apply >

< ! -- uav > : : = < Apply FunctionId = “& function; any-of” > < Function FunctionId = “(Function)” > /> < AttributeValue DataType = “(DataType)” > ( uav ) < /AttributeValue > < SubjectAttributeDesignator AttributeId = “(attributeid)” DataType = “(DataType)” > /> < /Apply >

< ! -- roprt > : : = < Apply FunctionId = “& function; any-of-any” > < Function FunctionId = “& function; String-match” > /> < SubjectAttributeDesignator AttributeId = “(attributeid)” DataType = “&XML; String” > /> < Apply FunctionId = “& function; String-bag” > { < AttributeValue DataType = “&XML; String” > “=” | “≠” < /AttributeValue > } + < /Ap-

Received 2008-04-15.

**Biography:** Ye Chunxiao (1973—), male, doctor, associate professor, yeec@cqu.edu.cn.

**Foundation item:** The National High Technology Research and Development Program of China (863 Program) (No. 2007AA01Z445).

**Citation:** Ye Chunxiao, Zhong Jiang, Feng Yong. Attribute-based access control policy specification language [J]. Journal of Southeast University (English Edition), 2008, 24(3): 260 – 263.

ply> </Apply>

$\langle ! \text{ -- oprt} \rangle :: = \langle \text{Apply FunctionId} = \text{"\& function; any-of-any"} \rangle \langle \text{Function FunctionId} = \text{"\& function; String-match"} \rangle \langle \text{SubjectAttributeDesignator AttributeId} = \text{"(attributeid)"} \rangle \langle \text{DataType} = \text{"\&XML; String"} \rangle \langle \text{Apply FunctionId} = \text{"\& function; String-bag"} \rangle \{ \langle \text{AttributeValue DataType} = \text{"\&XML; String"} \rangle \langle \text{"\<" | "\leq" | "=" | ">" | "\geq" | "\neq"} \rangle \langle \text{AttributeValue} \rangle \} + \langle \text{Apply} \rangle \langle \text{Apply} \rangle$

**Definition 4** uae, ae and RAE are defined in A-XACML as follows:

$\langle ! \text{ -- uae} \rangle :: = \langle ! \text{ -- ua} \rangle \langle ! \text{ -- roprt} \rangle \langle ! \text{ -- uav} \rangle$   
 $\langle ! \text{ -- ae} \rangle :: = \langle ! \text{ -- ua} \rangle \langle ! \text{ -- oprt} \rangle \langle ! \text{ -- uav} \rangle$   
 $\langle ! \text{ -- RAE} \rangle :: = \langle \text{Policy PolicyId} = \text{"(policyid)"} \rangle \text{RuleCombiningAlgId} = \text{"deny-override"} \{ \langle ! \text{ -- SRule} \rangle \} + \langle \text{Policy} \rangle$

where  $\langle ! \text{ -- SRule IN A-XACML} \rangle$  is defined as

$\langle \text{Rule RuleId} = \text{"(ruleid)"} \rangle \text{Effect} = \text{"(permit | deny)"} \rangle$   
 $\langle \text{Target} \rangle \langle \text{Subjects} \rangle \langle \text{AnySubject} \rangle \langle \text{Resources} \rangle \langle \text{AnyResource} \rangle \langle \text{Actions} \rangle \langle \text{AnyAction} \rangle \langle \text{Condition FunctionId} = \text{"\& function; And"} \rangle \langle ! \text{ -- uae} \rangle | \langle ! \text{ -- ae} \rangle \langle \text{Condition} \rangle \langle \text{Rule} \rangle$

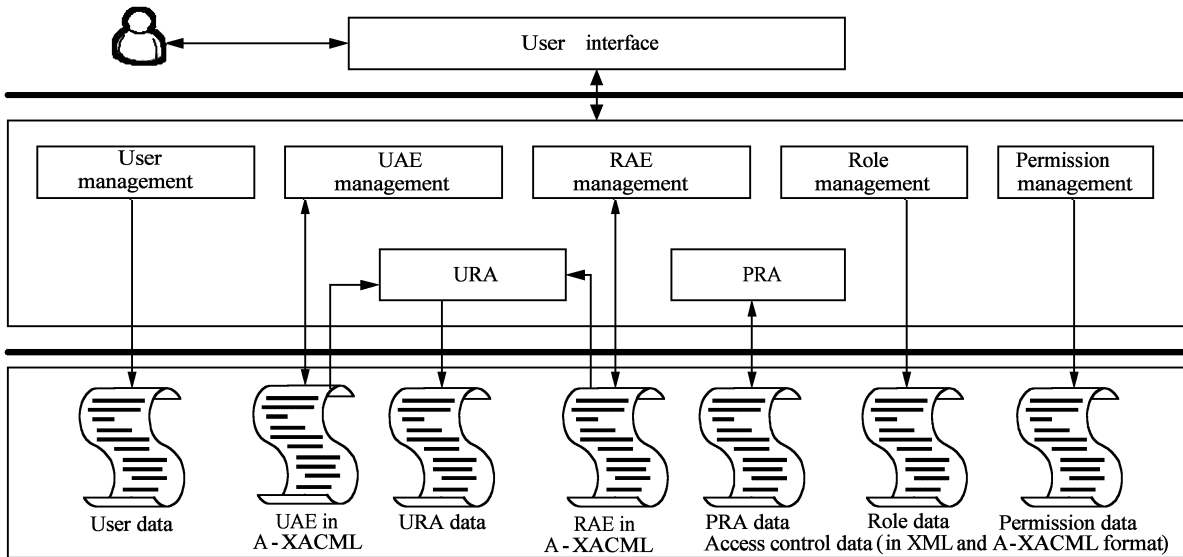
**Definition 5** UAE is defined as follows:

$\langle ! \text{ -- UAE} \rangle :: = \{ \langle ! \text{ -- User Attribute Expression} \rangle \} +$   
 where  $\langle ! \text{ -- User Attribute Expression} \rangle$  is defined as  
 $\langle \text{Subject} \rangle \langle \text{Attribute AttributeId} = \text{"(attributeid)"} \rangle \langle \text{AttributeValue DataType} = \text{"\&XML; String"} \rangle \langle \text{Attribute name} \rangle \langle \text{Attribute AttributeId} = \text{"(attributeid)"} \rangle \langle \text{Attribute Value DataType} = \text{"\&XML; String"} \rangle \langle \text{Attribute Operator} \rangle \langle \text{Attribute Value} \rangle \langle \text{Attribute AttributeId} = \text{"(attributeid)"} \rangle \langle \text{Attribute Value DataType} = \text{"(datatype)"} \rangle \langle \text{Attribute value} \rangle \langle \text{Attribute Value} \rangle \langle \text{Attribute} \rangle \langle \text{Subject} \rangle$

### 3 System Architecture and Case Study

Fig. 1 shows the main components of an access control system. In this system, access control policies are described as RAEs by using A-XACML and stored in A-XACML files. According to XACML standards, a user's UAE must be included in a request context which is compared to a role's RAE to judge whether this user-role assignment satisfies an access control policy or not.

Tab. 1 lists some roles and their RAEs. Supposing a user Liuz whose UAE is "QM\_experience = 2", here, we give a case to show how to describe a user's UAE and a role's RAE in A-XACML.



**Fig. 1** Main components of access control system

**Tab. 1** Role's requirements and RAEs

Role	Requirements and RAEs
DP (database programmer)	Requirements: three years of database programming experience and proficiency in SYBASE. RAE: database_experience >= 3 AND proficiency = Sybase.
JP (JAVA programmer)	Requirements: three years of java programming experience and proficiency in JAVA RAE: JAVA_experience >= 3 AND proficiency = JAVA
QM (quality manager)	Requirements: three years of quality management experience RAE: QM_experience >= 3

#### 3.1 User's UAE

The following as an example of a part of a request context includes user Liuz's UAE. The user's UAE is described in

the  $\langle \text{subject} \rangle$  element in lines 6 to 13, where the first  $\langle \text{Attribute} \rangle$  element denotes the ua of the UAE and the second denotes the roprt of the UAE and the third uav of the UAE. While in a user-role assignment, the first  $\langle \text{subject} \rangle$  in lines

2 to 5 must be matched to the  $\langle \text{subject} \rangle$  element of  $\langle \text{target} \rangle$  of the role's  $\langle \text{policyset} \rangle$ , while the  $\langle \text{attribute} \rangle$  elements of the second  $\langle \text{subject} \rangle$  element must be matched to corresponding  $\langle \text{apply} \rangle$  elements of  $\langle \text{condition} \rangle$  elements of role's  $\langle \text{policyset} \rangle$ s. Meanwhile,  $\langle \text{attribute} \rangle$  elements of the  $\langle \text{resource} \rangle$  and  $\langle \text{action} \rangle$  of the request context must be matched to corresponding  $\langle \text{action} \rangle$  and  $\langle \text{resource} \rangle$  elements of  $\langle \text{condition} \rangle$  elements of role's  $\langle \text{policyset} \rangle$ s. In this case, Liuz will be assigned to the role of QM.

```

1  $\langle \text{Request} \rangle$ 
2  $\langle \text{Subject} \rangle$ 
3    $\langle \text{Attribute AttributeId} = \text{"role-id"} \text{ DataType} = \text{"\#string"} \rangle$ 
4      $\langle \text{AttributeValue} \rangle \text{QM} \langle / \text{AttributeValue} \rangle$ 
5  $\langle / \text{Subject} \rangle$ 
6  $\langle \text{Subject} \rangle$ 
7    $\langle \text{Attribute AttributeId} = \text{"ua: ua-of-Liuz"} \text{ DataType} = \text{"\#string"} \rangle$ 
8      $\langle \text{AttributeValue} \rangle \text{QM\_experience} \langle / \text{AttributeValue} \rangle$ 
9    $\langle \text{Attribute AttributeId} = \text{"roprt: roprt-of-Liuz"} \text{ DataType} = \text{"\#string"} \rangle$ 
10      $\langle \text{AttributeValue} \rangle = \langle / \text{AttributeValue} \rangle$ 
11    $\langle \text{Attribute AttributeId} = \text{"uav: uav-of-Liuz"} \text{ DataType} = \text{"\#integer"} \rangle$ 
12      $\langle \text{AttributeValue} \rangle 2 \langle / \text{AttributeValue} \rangle$ 
13  $\langle / \text{Subject} \rangle$ 
14  $\langle \text{Resource} \rangle \dots \langle / \text{Resource} \rangle$ 
15  $\langle \text{Action} \rangle \dots \langle / \text{Action} \rangle$ 
16  $\langle / \text{Request} \rangle$ 

```

### 3.2 Role's RAE

In A-XACML, a role's RAEs are denoted by a  $\langle \text{policyset} \rangle$  with a single  $\langle \text{policy} \rangle$  element. A  $\langle \text{policy} \rangle$  may include many  $\langle \text{rule} \rangle$ s, each of which denotes an ae of a role. The value of the  $\langle \text{policy} \rangle$ 's rulecombiningalgid attribute is "permit-overrides" for the relationships among these RAE is "AND". A role's operations and objects are denoted by an  $\langle \text{action} \rangle$  element and a  $\langle \text{resource} \rangle$  element of a  $\langle \text{rule} \rangle$ , respectively. A pae of a role's AE is denoted by a  $\langle \text{condition} \rangle$  element of a  $\langle \text{rule} \rangle$ , where its ua, poprt and uav are denoted by an  $\langle \text{apply} \rangle$  element respectively. The following is an example of a roles' RAE. In this example, the role QM's RAE " $\text{QM\_experience} = 3$ " is described in a  $\langle \text{condition} \rangle$  element. The oprt " $=$ " is described as " $=$ " in line 12, but the function "integer-greater-than-or-equal" in line 14 means the corresponding uav of a user's UAE must " $= 3$ ".

```

1  $\langle \text{PolicySet PolicySetId} = \text{"QM"} \text{ PolicyCombiningAlgId} = \text{"policy-combine: permit-overrides"} \rangle$ 
2  $\langle \text{Condition FunctionId} = \text{"function: and"} \rangle$ 
3    $\langle \text{Apply FunctionId} = \text{"function: string-equal"} \rangle$ 
4      $\langle \text{Apply FunctionId} = \text{"function: string-one-and-only"} \rangle$ 
5        $\langle \text{SubjectAttributeDesignator AttributeId} = \text{"ua: ua-of-qm-experience"} \rangle$ 
6      $\langle / \text{Apply} \rangle$ 
7      $\langle \text{AttributeValue DataType} = \text{"\#string"} \rangle \text{QM\_experi-}$ 

```

```

ence  $\langle / \text{AttributeValue} \rangle$ 
8    $\langle / \text{Apply} \rangle$ 
9    $\langle \text{Apply FunctionId} = \text{"function: string-one-and-only"} \rangle$ 
10      $\langle \text{SubjectAttributeDesignator AttributeId} = \text{"roprt-of-qm-experience"} \rangle$ 
11    $\langle / \text{Apply} \rangle$ 
12    $\langle \text{AttributeValue DataType} = \text{"\#string"} \rangle = \langle / \text{AttributeValue} \rangle$ 
13    $\langle / \text{Apply} \rangle$ 
14    $\langle \text{Apply FunctionId} = \text{"function: integer-greater-than-or-equal"} \rangle$ 
15      $\langle \text{Apply FunctionId} = \text{"function: string-one-and-only"} \rangle$ 
16        $\langle \text{SubjectAttributeDesignator AttributeId} = \text{"uav: uav-of-qm-experience"} \rangle$ 
17      $\langle / \text{Apply} \rangle$ 
18      $\langle \text{AttributeValue DataType} = \text{"\#string"} \rangle 3 \langle / \text{AttributeValue} \rangle$ 
19    $\langle / \text{Apply} \rangle$ 
20  $\langle / \text{Condition} \rangle$ 
21  $\vdots$ 
22  $\langle / \text{PolicySet} \rangle$ 

```

## 4 Conclusion

In this paper, we first use user and role attribute expressions to describe access control policy. Secondly, we propose an A-XACML language, which is an extension of the existing XACML, to describe user and role attribute expressions, such as UAE and RAE. We also give a case to show how A-XACML works in a URA. The case shows that A-XACML can describe access control policy in a more powerful and simpler way. We believe that to specify and enforce more access control policy with XACML is an interesting topic for future study.

## References

- [1] Sandhu Ravi, Coyne Edward, Feinstein Hal, et al. Role-based access control models [J]. *IEEE Computer*, 1996, **29** (2): 38–47.
- [2] Bhatti R, Bertino E, Ghafoor A, et al. XML-based specification for web-services document security [J]. *IEEE Computer*, 2004, **37**(4): 41–49.
- [3] Joshi J, Bhatti R, Bertino E, et al. Access-control language for multi-domain environments [J]. *IEEE Internet Computing*, 2004, **8**(6): 40–50.
- [4] Godik Simon, Moses Tim, Anderson Anne, et al. OASIS extensible access control markup language (XACML) [EB/OL]. (2005-12-02) [2008-02-12]. <http://www.oasis-open.org/committees/xacml/>.
- [5] Toktar E, Jamhour E, Maziero C. RSVP policy control using XACML [C]//*Proc of POLICY'04*. New York: IEEE Computer Society Press, 2004: 87–98.
- [6] Al-Kahtani Mohammad Abdullah. A family of models for rule-based user-role assignment [D]. Fairfax: Department of Computer Science of George Mason University, 2003.

# 基于属性的访问控制策略描述语言

叶春晓 钟 将 冯 永

(重庆大学计算机学院, 重庆 400044)

**摘要:**首先提出了基于属性的访问控制策略,该方法利用用户和角色属性表达式来描述访问控制策略.然后,提出了扩展的 XACML(扩展访问控制标记语言)策略描述语言 A-XACML. A-XACML 可以简单、灵活地表达各种应用环境中的访问控制策略,尤其是基于属性的访问控制策略.该语言及其框架通过数据类型、函数和逻辑组合来定义简单或复杂的访问控制策略.最后,给出了利用属性表达式和 A-XACML 来实现用户-角色指派的系统架构和应用实例.该实例表明属性表达式和 A-XACML 能够灵活简单地描述和实施复杂的访问控制策略.

**关键词:**基于角色的访问控制;策略;XML;XACML

**中图分类号:**TP309.2