Management Advantages of Object Classification in Role-Based Access Control (RBAC)

Mohammad Jafari and Mohammad Fathian

Department of Information Technology, Faculty of Industrial Engineering, University of Science and Technology (IUST), Narmak, Tehran, Iran
{m_jafari, fathian}@iust.ac.ir

Abstract. This paper investigates the advantages of enabling object classification in role-based access control (RBAC). First, it is shown how the merits of the RBAC models can be ascribed to its using of abstraction and state of dependencies. Following same arguments, it is shown how inclusion of object classification will ameliorate dependencies and abstractions in the model. The discussion contains examining seven criteria to compare object-classification-enabled RBAC with plain RBAC and trivial-permission-assignment models, in order to show the advantages of object classification in a more formal manner. The criteria are: number and complexity of decisions, change management cost, risk of errors, policy portability and reuse, enforcement and compliance, support for traditional information classification policies, and object grouping and management support.

Keywords: Access Control, Role-Based Access Control (RBAC), Object Classification.

1 Introduction

The family of RBAC models is very well studied in the literature; borders have been clarified by introducing reference models [20], and finally, it has been codified in form of a standard [1]. Many extensions have been proposed to RBAC in order to increase its power and expressiveness. This paper will focus on object classification as one of such extensions and argue how it can improve its management efficiency.

Many contributors have glimpsed the idea of object classification during their discussion of RBAC. Sandhu mentions the concept of "object attributes" as a means of grouping objects, the same way as roles categorize subjects; though he doubts whether this idea fits in the scope of RBAC [19]. Later, he hints at the idea of "generic permission" as a special form of permission applied only to one group of objects; nonetheless, he neglects the concept as being a matter of implementation [20]. "Team-based access control" is another scheme which limits access rights of users to their team's resources [22]. It can be viewed as an effort for object classification. In this model, objects are grouped into generic entities named "object types" and the permissions of each role are expressed in form of rights to access these "object types" rather than objects themselves. The notion of "role templates" proposed in [11] is an effort to restrict the privileges of a role to certain kind of objects in order to make

I. Cervesato (Ed.): ASIAN 2007, LNCS 4846, pp. 95 – 110, 2007.

content-based access control possible. Roles templates, special "parameterized roles", are actually a means to classifying objects. Objects are classified into categories, which are then used as parameters to role templates, in order to limit the authority of the role to a single category of objects. This notion is used in [12] as a basis to introduce the concept of "object-specific role", a special kind of role the capabilities of which is restricted to a certain group of objects. In other words, this work suggests manipulating the meaning of role, in order to make object classification possible.

The most significant work on object classification however, seems to be done by Covington et al. in [3]. The notion of "object roles" in their "generalized RBAC" is the most evident effort to empower RBAC with object classification, and is similar to the approach of this paper from a conceptual point of view. Recently, Junghwa in [24] proposed a formalization of object classification together with support of object hierarchies and provided some reason in favor of adding this concept to RBAC.

Although rarely noticed in the mainstream of RBAC-related literature, many implementers of RBAC have realized the importance of object classification and include mechanisms to support it. Hence, the notion of object classification is no new idea in the world of implementation. There are often a large number of objects in real systems and defining access rights regarding every single object is impractical [16]. Classification of files in form of directories and applying access rights to the whole directory is one typical example. Using DTD schema as a categorizing mechanism for XML documents in [4, 5] can serve as another instance.

This paper starts with establishing a conceptual basis for measuring the management efficiency of an access control model by focusing on the notion of "dependencies". Three typical models are then considered as the center of discussions: "TPA model", "Plain RBAC", and "object-classification-enabled RBAC", coded as TPA, P-RBAC, and OC-RBAC respectively. On the basis of dependencies and abstractions, it is shown how object classification can bring about many management advantages compared with P-RBAC and TPA models. These insights are then formulated in form of Omicron notation (O(n)). Taken together, the main contribution of this paper is to provide arguments in favor of enabling object classification in RBAC and formulating them.

The remainder of this work proceeds as follows: In section 1.1 an overview of RBAC model is presented in which particular attentions is paid to the state of "dependencies" between the entities of the model. On this ground, some shortcomings believed to exist in the P-RBAC are discussed in section1.2. Section 2 outlines object classification in its simplest form which is then formalized in 2.1 by emulating the definitions of RBAC. Section 3 dwells upon the advantages of OC-RBAC through examination of the seven criteria. Section 4 is where the paper concludes with a summary and probable future works.

1.1 RBAC Review

From a managerial point of view, one of the main points underlying RBAC is separating subjects from their access permission, using an extra layer of abstraction, named "role". The keyword here is "abstraction" which is a well-known concept in

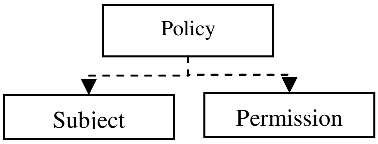

Fig. 1. Dependencies of the access control policy in TPA

system design. From this point of view, RBAC model contrasts TPA models (such as access matrix), in which subjects' permissions are directly assigned to them.

In TPA models, access control policy is stated in form of (subject, permission) pairs, and hence, each of its entries contains a reference to a subject and a permission. This dependency is the root of many problems, as will be discussed later. Access control lists are one of the most well-known examples of using such a model that suffer from many managerial deficiencies. One of the most important contributions of RBAC is believed to be improvement of their manageability [8].

RBAC eliminates the direct relationship between subjects and permissions by setting up the "role" entity which mediates between the two and removes the coupling of policy to permissions. In this model, access control policy can be divided into two components: one component decides the roles of each subject and the other component specifies the access rights of each role. The two components can be expressed in form of subject-role and role-permission pairs respectively. These two components are henceforth called "major" and "minor" components of the RBAC policy to accentuate their cost and importance which will be discussed in section 3.1. (figure 2.a).

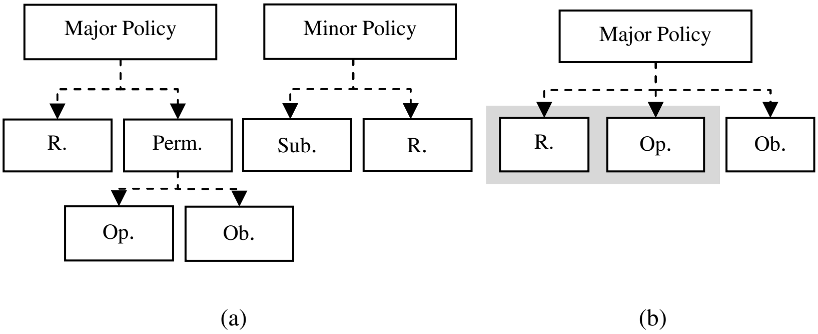

Fig. 2. a. Major and minor access control policies with their dependencies. b. Detailed dependencies of the major access control policy in RBAC model. The permission entity is replaced by its constituents, namely objects and operations. The shading denotes less-frequently changing entities.

1.2 Absence of Object Classification in P-RBAC

The "permission" entity deserves more elaboration as it is a key entity in any access control model. Permission is a general term that refers to the right of doing some unit of work in a system [10]. For the sake of simplicity, we will skip complex forms of permissions and assume that permission is composed of an operation exercised on an object. Therefore permissions embody the relation between the objects and operations (figure 2.a). This reveals the dependency of permissions to operations as well as objects, and consequently, the dependency of major access control policy to objects and operations in P-RBAC-based systems (figure 2.b).

Normally, the set of operations is constant across all similar systems, because the set of possible operations is related to the essence of a system [1]. A similar argument holds about roles. Roles are the same across similar systems, because they correspond to the nature of the system. It has also been argued that roles must be engineered in such a manner that they remain stable even against business restructuring [18]. For example, the set of operations (credit, debit, etc.) and roles (clerk, accountant, manager, etc.) are similar among all banking systems; contrary to the set of objects which is dependant to a particular instance of a system. Any banking system has its own set of objects (particular accounts, bills, etc.), even in different departments of a same company. This persistent nature of roles and operations is depicted by the shading in figure 2.b.

The dependency of major access control policy to objects implies that the role-permission decision is utterly an organization-dependent practice. Despite the abstract and system-independent nature of roles, permissions, and hence the whole policy, are dependent to objects. This means that the major access control policy is dependent to one particular system, and implies that the same process of role-permission assignment must be reiterated even for most similar organizations.

Moreover, in the implementation level, the major access control policy will experience a tough coupling to object names (as it is apparent in functional specifications of RBAC models in [9] and [1]). Therefore, any changes in the set of objects of the system, such as adding or renaming objects, will obligate an update to the major access control policy. This is not suitable, as the major access control policy is a very sensitive piece of information, and it might be desirable to store it as read-only.

Lack of management facilities for objects at the model level is another fact that highlights absence of object classification in P-RBAC. One of the most important advantages of RBAC is its administrative power of managing users in form of roles [6, 7, 9]. However, such a power is missing for objects, at least in the model level. In systems based on P-RBAC, objects, even if they are quite similar, are treated separately, and there is no abstraction support in the model for grouping them. Proposing concepts such as "object attributes" in [19], or "generic permissions" in [20] are efforts to solve this problem.

These problems can be traced back to the imbalanced state of the dependencies in the RBAC model. The following section will try to show how object classification can solve these problems, by enhancing the state of dependencies in the model.

2 Proposed Object Classification Scheme

Object classification is realized by declaring a new entity named "category". Other names such as "object role" or "object class" have been proposed for similar concepts elsewhere in the literature [3, 16]. The abstraction of "category" serves the same functionality to objects as the abstraction of "role" does to subjects. A many-to-many relation is defined between objects and categories by which objects can be grouped and classified from several different points of views. Access rights are granted to roles in by using categories in the major access control policy. A subject is authorized to access an object iff at least one of its roles is allowed to access one of the categories assigned to that object.

In this scheme, major access control policy would no more depend on the objects themselves, but rather on object categories. Permissions will now stay on a higher level and involve operations on categories rather than system-specific objects. Consequently, the dependencies of the major access control policy will be refined as shown in figure 3.b. As depicted in figure 3.b, major policy is no more depending upon any frequently-changing entities. Since the category of each object should be determined, an extra component will appear in the minor part of access control policy in order to decide categories of objects (figure 3.a).

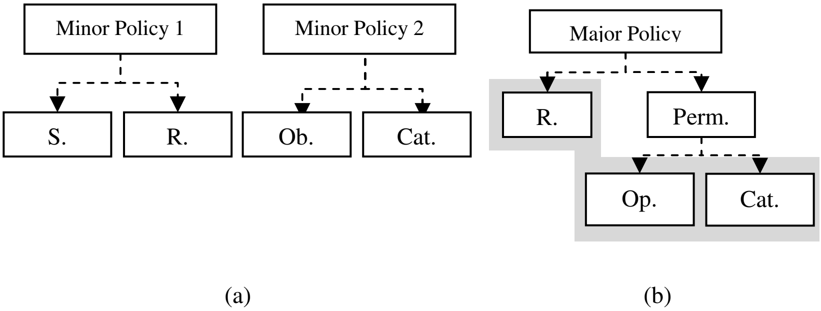

Fig. 3. a. Minor access control policies and their dependencies when object classification is involved. b. Major access control policy dependencies when object classification is involved.

From a managerial point of view, three steps are needed to be taken in order to establish the major access control policy in OC-RBAC: Role engineering, category engineering and policy decision. The role engineering and policy establishment processes are similar to P-RBAC. The category engineering may be associated with the process of "asset classification" which is a major domain in information security management standards [14, 15]. Actually the asset classification practice can be realized in form of category engineering.

2.1 Formal Definition

Here, a formal definition of OC-RBAC is presented. This definition obviously resembles the definitions of RBAC. Some details of the original RBAC model (e.g. role activation) are intentionally eliminated for the sake of brevity. Bringing those concepts back to the model is straightforward. SUBS, ROLES, OPS, CATS, and OBS are the sets of subjects, roles, operations, categories and objects respectively.

PRMS = (OPS×CATS) which denotes the set of all permissions. It is noteworthy that not all pairs in this set are semantically meaningful because not every operation is valid for all categories. Some operations might be meaningful only for some particular categories of objects. For example, in a file system, "read" and "write" operations are applicable to all kinds of files while the "execute" operation is applicable only to a specific file category, called executables. This observation opens the way to defining integrity constraints on the set of permissions.

SRA $\subseteq$ SUBS × ROLES, a many-to-many mapping between subjects and roles which denotes the assigned roles of each subject.

OCA $\subseteq$ OBS × CATS, a many-to-many mapping between objects and categories which denotes the assigned categories of each object.

PRA $\subseteq$ PRMS × ROLES, a many-to-many mapping between permissions and roles which denotes the assigned permissions of each role.

asnd_roles: (s:SUBS)→ ROLES the mapping of subject *s* onto a set of roles which denotes assigned roles of a subject. Formally:

asnd_roles (s) = $\{r\in \mathrm{ROLES} | (s,r) \in \mathrm{SRA}\}$

asnd_cats: (o:OBS)→ CATS the mapping of object *o* onto a set of categories which denotes assigned categories of an object. Formally:

asnd_cats (o) = $\{c\in \mathrm{CATS} | (o,c) \in \mathrm{OCA}\}$

asnd_prms: (r:ROLES)→ PRMS the mapping of role *r* onto a set of permissions which denotes the permissions of a role. Formally:

asnd_prms (r) = $\{p\in \mathrm{PRMS} | (p,r) \in \mathrm{PRA}\}$

acc: SUBS×OPS×OBS→Boolean, which denotes whether a subject is authorized to perform an operation on an object. acc(s,op,o)=TRUE if the subject *s* is allowed to perform operation *op* on object *o* and FALSE otherwise.

Modified object access authorization property: A subject *s* can perform an operation *op* on object *o* only if there exists a role *r* that is included in the subject's roles set and there exists a permission in *r*'s assigned permissions set that authorizes performing operation *op* on one of the categories containing object *o*. Namely:

$$\begin{gathered}\text{access }(s,op,o) \Rightarrow \\ \exists r\in \mathrm{ROLES},\ \exists c\in \mathrm{CATS},\ \exists p\in \mathrm{PRMS} \\ r \in \text{asnd\_roles }(s) \wedge p \in \text{asnd\_perms }(r) \wedge (op,c)\in p \wedge c\in \text{asnd\_cats }(o)\end{gathered}$$

3 Discussion

Object classification is very similar to roles-subjects assignment and hence its advantages can be intuitively sensed through comparison. The concept of category improves the state of dependencies in the P-RBAC model, the same way the concept of role did to TPA model. This section will enumerate the advantages of OC-RBAC over the P-RBAC and TPA. Seven criteria are considered for the comparison of models and the results are summarized in Table 1. The criteria are: number and complexity of decisions, change management costs, policy portability and reuse, risk of errors, ease of enforcement and compliance, ease of applying information classification policies, and support from model for object grouping and management.

Table 1. Comparing management features of the three models for access control. M/m: manager-level/operator-level complexity; I/i: high/low impact (manager/operator); P/p: higher/lower probability.

	TPA	P-RBAC	OC-RBAC
Number and Complexity of Decisions	M.O(n^2)	M.O(n) + m.O(n)	M.O(1) + m.O(n)
Change Management Cost (Detailed in table 2)	very poor	good	better
Risk of Errors (Error Likelihood × Impact)	I.p.O(n^2)	I.p.O(n) + i.P.O(n)	i.P.O(n) + I.p.O(1)
Policy Portability and Reuse	None	M.O(n) + m.O(*n*)	m.O(*n*)
Enforcement and Compliance	None	Manual	Automated
Support for Traditional Information Classification Policies	None	Complex	Trivial
Object Grouping and Management Support	Implementation-level	Implementation-level	Direct Support from Model

3.1 Number and Complexity of Decisions

Management complexity is an important criterion for evaluating the models in question. More complex models need more management resources and therefore lead to more management costs. Moreover, complexity is the root cause of many management errors and also complicates evaluation. Thus, reducing management complexity can be considered as an advantage. We will assume two indicators for the complexity of an access control model: the number of decisions to be made in order to establish the access control policy, and the complexity of each decision. The number of decisions to be made in the process of policy establishment is a good indicator of the management complexity of an access control system because larger number of decisions means utilizing more management resources and higher probability of unintentional mistakes [8]. But counting the number of decisions is not sufficient as different decisions involve different costs. Decisions may be so simple that can be made by an operator or so complicated that require involvement of the board of directors. Noticing the difference between major and minor components of the access control policy, it can be observed that different costs are burdened for decisions in each of the two components. The major access control policy wich involves role-permission decisions needs far more elaboration than the subject-role decisions of minor access control policy. Since roles are often correspond to organizational positions, deciding the role of a subject is a daily task that can be done by an ordinary operator. Even, there have been some efforts to automate such task by using rule-based mechanisms [23]. For instance, it is trivial to realize who the secretary or server administrator is, and therefore, assigning these roles to the corresponding subjects is straightforward. On the contrary, deciding about the access rights of a role in an organization is a complex job usually done by managers and security engineers, and may also need to be examined against higher-level security policies of a system. For instance, when a new employee enters in a company, say in clerk in a bank office, his/her organizational position is usually straightforward and an ordinary operator can enter this information in the system. But when a new position appears in the organization, such as a new "IT manager" position in a bank office, careful study should be undertaken to detail the permissions of this new role. Besides, new positions usually appear as a consequence of some organizational change which is supposed to happen very rarely; while leaving or joining an organization, or changes in organizational positions are regular events which may occur even daily in large systems. This distinction will play a major role in estimating management cost of the decisions in each of these two types of policies. Thus, different costs must be assumed for the sorts of decisions in the two components of the access control policy. We will use m for the cost of a minor policy decisions and M for a decision in major access control policy.

Multiplying total number of decisions of each type by their cost, the total cost imposed by each model can be calculated as an indicator of its complexity. The sets S, Op, O, R and C which are the set of all subjects, operations, objects, roles and categories respectively are important parameters in this calculation. T_i and d_i are the total cost and the total number of decisions of the each model respectively.

In the TPA model depicted in figure 1, the policy is determined by deciding whether to permit each of the triplets of the form (s, op, o) in which $s \in S$, $op \in Op$ and $o \in O$. So, the total number of decisions can be calculated as $d_1 = |S|.|Op|.|O|$ in which $|S|$, $|Op|$ and $|O|$ are the number of elements in the corresponding sets. As discussed in section 1.1, normally there are constant number of operations in a system. This claim is trivial and also confirmed by practical case studies [22]. So, $d_1 = |S|.|O|.\mathrm{const}$. If n is assumed to be the maximum of $|S|$ and $|O|$, the total number of decisions in the TPA model follows:

$$d_1 = O(n).O(n).O(1) = O(n^2).$$

All of the decisions in TPA are major because they involve deciding the access rights of a subject. Therefore the total cost for the TPA is figured out as:

$$T_1 = M.O(n^2). \tag{1}$$

In the P-RBAC model there are two policy components. The minor policy is determined by deciding the roles of each subject. There are $|R|$ roles in the system and $|S|$ subjects, and it should be determined whether each subject is the member of each role. Therefore, for each pair (s,r) in which $s \in S$ and $r \in R$, there is a binary decision to determine whether the subject s is a member of role r. Thus, $d_{2\,(Minor)} = |S|.|R|$.

The major policy is determined by deciding whether to permit each of the triplets (r, op, o) in which $r \in R$, $op \in Op$ and $o \in O$. Consequently, $d_{2\,(Major)} = |R|.|Op|.|O|$ and the total number of decisions can be summed up as follows:

$$d_2 = d_{2\,(Major)} + d_{2\,(Minor)} = |S|.|R| + |R|.|Op|.|O|.$$

The number of operations is constant as discussed before. The number of roles in the system can also be assumed as constant because it is insignificant compared to the number of subjects and objects and it does not grow significantly as the system gets larger. This was discussed in section 1.1 and is also shown to be true in practical case studies as in [22] and [6]. Thus, $d_2 = |S|.const + |O|.const$. Multiplying each term by its cost, the total management cost of the P-RBAC model can be figured out as follows. Again n is assumed to be the maximum of $|S|$ and $|O|$:

$$T_2 = M.O(n) + m.O(n). \tag{2}$$

The access control policy in OC-RBAC model is composed of one major and two minor components. The first minor component of the policy pertains to determining each subject's roles and is similar to the one in the P-RBAC model; therefore $d_{3\,(Minor\,1)} = |S|.|R|$. Likewise, the other minor component of the policy involves assigning appropriate categories to each object; thus $d_{3\,(Minor\,2)} = |O|.|C|$.

The major policy is similar to that of P-RBAC with the set of objects replaced by the set of categories; therefore $d_{3\,(Major)} = |R|.|Op|.|C|$. As argued before, the number of operations in a system is constant. The number of categories is also constant with similar reasons as given for roles. Accordingly, the total number of decisions is summed up as follows:

$$d_3 = d_{3\,(Major)} + d_{3\,(Minor\,1)} + d_{3\,(Minor\,2)} = |S|.|R| + |O|.|C| + |R|.|Op|.|C|$$
$$= |S|.const + |O|.const + const.$$

By assuming n=max $(|S|,|O|)$, and multiplying costs total cost of OC-RBAC is:

$$T_3 = m.O(n) + m.O(n) + M\,.const = m.O(n) + M.O(1) \tag{3}$$

Considering equations 1 through 3, one can vividly observe an improvement in the complexity of models. As the system grows, more objects and subjects enter the system and the value of n increases. The growth of complexity in the TPA model is of quadratic order while the complexity of P-RBAC grows linearly. In P-RBAC models, the growth function is composed of a term with the factor of M (administrative cost) as well as a term with factor of m (operator cost). This implies that the growth of complexity is endured by both managers and operators. However, when object classification is involved, the growth function has only a term with the factor of m (operator costs) and the complexity is shouldered only by operators. Accordingly, object classification can lead to significant reduction in the management complexity of the access control system.

3.2 Change Management Cost

Change can occur in many forms to an access control policy and ease of managing change is a major criterion for evaluating access control models. Here, some typical forms of change are discussed and the capabilities of each model in managing them are compared. A summary of this comparison is depicted in Table 2.

Table 2. Cost of managing five typical sorts of change in the models under discussion. (M: manager-level complexity; m: operator-level complexity).

Change Type	TPA	P-RBAC	OC-RBAC
subject access rights	M.O(n)	m.O(1)	m.O(1)
role access rights	M.O(n^2)	M.O(n)	M.O(1)
object access permissions	M.O(n)	M.O(1)	m.O(1)
category permissions	M.O(n^2)	M.O(n)	M.O(1)
total change	M.O(n^2)	m.O(n)+ M.O(n)	m.O(n) + M.O(1)

Subject Access Rights: In the TPA model, changing the access rights of a subject usually involves reviewing all of its rights to access every single object in the system. Given that the number of objects is n, this involves O(n) operations. All of these operations are administrative since they involve a decision in major access control policy; thus they are weighted with M and the total cost is M.O(n).

In both plain and OC-RBAC, normal changes in a subject's access right mean a change in that subject's role and random changes in subjects' rights are not supposed to happen on a regular basis. Actually, if changes to subject's rights cannot be interpreted into changes in its role, then the role engineering in the system is flawed and "new" or "modified" roles need to be introduced. Consequently, we can safely presume that any changes to subject's access rights is in form of changes in its role which involve only O(1) operation. These operations are related to the minor access control policy and cost m, thereby leading to the total cost of m.O(1).

Role Access Rights: A major change in the access rights of a group of subjects may imply a change in the access rights of a role. In the TPA model, there is no support for roles; consequently this kind of change will lead to reviewing access rights of a number of subjects. As explained before, the cost of changing the access rights of a single subject is M.O(n). Accordingly, changing the access rights of a group of subjects is M.O(n^2), because this groups may contain as many as O(n) subjects.

In P-RBAC, this kind of change can be handled by reviewing a role's rights to access each object in the system which takes O(n) operations, since the number of objects is O(n). Since these operations belong to major access control policy, they cost M and thus, the total cost is M.O(n).

In the OC-RBAC model such changes can be handled by reviewing the particular role's rights to access each category of objects. As the number of categories is constant, this involves only O(1) major operations which leads to a cost of M.O(1) in total.

Object Access Permissions: This occurs when an object is at the focus of the change. For example, when the security label of a document is changed from "top secret" to "secret", an object-centric change takes place. In TPA model, handling such a change involves reviewing every subject's rights to access the object in question. This takes O(n) operations as total number of subjects is O(n). All of these operations are administrative the cost of which is M, thereby leading to total cost of M.O(n).

In P-RBAC, there is no need to examine the access rights of every single subject since roles can be examined instead. Hence, this kind of change takes only O(1) operations as the number of roles in the system is limited. However, these operations are all administrative and cost M because they involve a decision about the access rights of roles and belong to the major access control policy. This leads to total cost of M.O(1).

If object classification is available, there are well-engineered categories that group objects together in a logical manner. For this reason, it can be assumed that permissions relating to an object do not change arbitrarily, but rather in form of a change in its set of assigned categories. Changing the categories of an object involves a single decision in the minor component of the access control policy, and hence costs m, which leads to total cost of m.O(1).

Category Access Permission: Although rarely, there are times when altering access rights of a whole category of objects is necessary. An example of such change is when the roles that can access a confidential document need to be changed. In TPA model, this case resembles the case of changing a subject's access rights, which involves examining all subjects' rights to access each object in the system, leading to a cost of M.O(n^2).

In P-RBAC model there is no support for categories; therefore in such a case, each role's rights to access corresponding objects must be reexamined. This involves O(n) decisions for each of the roles in the system which leads to a total of O(n).O(1) operations. Since these operations are administrative and cost M, the total cost of this kind of change for this model is M.O(n).

If object classification is enabled, changing the permissions of a category involves reinspection of the each role's rights to access that particular category. This takes only O(1) administrative operations leading to M.O(1) total cost.

Total Change in Some Area: There may be times when a major revision of the access control policy is required which involves a number of object and subjects from different roles and categories. This kind of change is so severe that no role or categories can be preserved and a complete reengineering in needed in that area of the system. In such cases, that particular subset of the system can be assumed as a single system which needs policy establishment from scratch. The cost of this total reengineering is similar to the cost of complete policy establishment process that was calculated in section 3.1.

3.3 Risk of Errors

The total risk of errors involved in the management process is another criterion for comparing the three models under discussion. The risk of error is the product of error probability and error impact. Since major management decisions are made through more elaboration and by allocating more resources (such as committees, double checking, formal acceptance, etc.) the probability of making an error can be assumed to be lower than operator decisions; therefore different probabilities are assumed for administrative and operator errors which are denoted by p and P respectively. On the other hand, error in a management decision has a more profound impact than an error made by an operator; thus, different impact factors are assumed for these two kinds of decisions which are denoted by I and i respectively. Accordingly, each management decision involves a risk of $I.p$ (low-probability, but high-impact) while operator decisions have a risk of $i.P$ (low-impact, but high probability).

In the TPA model, there are O(n^2) management decisions each of which incurs a risk of $I.p$, therefore the total risk of errors in this model is $I.p$.O(n^2). In P-RBAC however, there are O(n) management decisions as well as O(n) operator decisions, leading to a total risk of $I.p$.O(n) + $i.P$.O(n). In the OC-RBAC model, there are O(n) operator decisions and O(1) management decisions; therefore the total risk is $i.P$.O(n) + $I.p$.O(1). Comparing the three figures, the advantage of P-RBAC over TPA, and similarly, the advantage of OC-RBAC over P-RBAC is obvious.

3.4 Policy Portability

Policy portability can be of value to many organizations. Porting the access control policy to branch offices and subsidiaries brings about management and financial advantages as well as policy consistency. Moreover, similar organizations that share same sets of roles, categories and operations can benefit from this capability by collaborating to develop a shared access control policy and thus economize in security costs.

The TPA model has no provisions for such a notion as the access control policy is tightly system-dependent. P-RBAC however, has facilitated policy portability to some extent by abstracting subjects in form of more general entities namely roles. Similar organization sharing a same set of roles can use the same major access control policy if they modify the permissions to include their own objects. Therefore policy portability is possible provided that some manual modifications are applied. These modifications comprise revising the major access control policy to take objects of the new system into account. There are O(1) roles and O(n) objects in the new system, and revising the major access control policy requires deciding the rights of each of the roles to access each object, which needs a sum of O(n) management decisions costing M. In order to have a complete access control policy, the minor component must also be established. This requires O(n) operator decisions for determining the members of each role. The total cost of porting a P-RBAC policy to a new system is thereby M.O(n) + m.O(n).

When object classification is available, since the major access control policy does not rely on any system-specific entities (objects or subjects), it is general enough to be ported to similar systems automatically and without manual modifications. Roles, operations and categories stay nearly the same across all organizations of the same type because they are related to the essence of a system rather than a particular instance. Results from case studies do not oppose this presumption [21, 6]. Accordingly, since the major policy needs no change, it can be ported without modification and the adopting system only needs to develop its own minor access control policies in order to assign local subject and objects to existing roles and categories respectively. As discussed before, this incurs O(n) decision of operator cost, leading to the total cost of m.O(n).

As a very simplified example, a software development environment can be assumed in which there are some software managers (SM), a number of developers (D) and several quality managers (QM). Objects in this environment can be grouped into source code (SC), test case (TC), management document (MD), and developer documents (DD). Typical operations can be recounted as create, read, modify, and execute the latter of which is only applicable to "source code" and "test case". Setting all forms of inheritance aside, a very simple major access control policy can look like as shown in table 3.

The access control policy depicted in the foregoing example is general enough to be adopted by several software projects and can serve as a standard policy of a company. Each project only needs to specify the members of roles and categories (minor components of the policy) trivially in order to have a complete access control policy. In this manner, the major access control policy can be ported and reused many times across similar systems.

Table 3. A simplified instance of major access control policy for a software project; the policy is general enough to be adopted by several projects. (R:read; M:modify; C:create; E:execute).

	Source Code	Test Case	Management Documents	Developers Document
Project Manager	R	E	C/R/M	R
Quality Manager	R	C/E	R/M	R/M
Developer	C/R/M/E	E	-	C/R/M

3.5 Enforcement and Compliance

System-independence has a very significant advantage for policy establishment authorities like government agencies and national or international standard bodies. In the medical arena as an example, there can be a unified set of roles, operations and categories that holds for any health care organization. Therefore a regulatory body can establish a general policy for all of the similar organizations in one field and enforce it. Consequently, national or international access control policies serving as unifying standards are possible. Enforcement of such policies can be easily automated by requiring use of a particular major access control policy. The subordinate systems would be required to use a standard major access control policy and hence, make sure they comply with the standard. Automated policy enforcement and compliance checking result from the abstract nature of the major access control policy which is the direct outcome of using object classification. Such facility is neither present in the TPA model nor in P-RBAC. However, as P-RBAC policies contain some level of abstraction, enforcing an RBAC policy is possible in a manual manner and by human intervention. The policy portability and the ability to express global policies can be considered as being among the most important advantages of equipping RBAC with object classification. These features can be seen as a further realization of the original goals of RBAC for elevating the access control policy from a matter of implementation to a high-level organizational and even inter-organizational issue as noted in [19].

3.6 Support for Traditional Information Classification Policies

Information classification is one of the traditional origins of access control and is still needed by current security applications. This can be in form of vertical information

Table 4. Expressing Bell-LaPadula security policy by using object classification

Category Role	Top-Secret	Secret	Confidential	Unclassified	…
Top-Secret	read/write	read	read	read	
Secret	write	read/write	read	read	
Confidential	write	write	read/write	read	
Unclassified	write	write	write	read/write	
…					

classification of military systems or horizontal classification which is more commonly used by civilian organizations. These kinds of policies cannot be expressed in TPA model in a systematic manner due to lack of abstraction. In P-RBAC, expressing such policies is a complex job which involves complicated schemes [17]. However, using object classification, expressing such policies is straightforward. Table 4 depicts a simple major policy similar to the well-known Bell-LaPadula [2] policy.

3.7 Object Management and Grouping Support

One of the obvious advantages of categorizing objects is the ability to manage them more systematically through grouping. Beyond trivial management advantages that result from hierarchical grouping of objects, this can prevent inconsistency in access control policies caused by unintentional mistakes. Furthermore, it can be helpful in eliminating redundancy at the implementation level. These benefits have encouraged RBAC implementers to include some form of object classification in their product, although there is no direct support for such concepts in the model. Enabling object classification in the model-level acts as a unifying mechanism for all object classification implementations.

4 Conclusion

This paper showed how the merits of role-based access control model can be traced back to its state of dependencies and abstractions and by following this interpretation it formalized the benefits of object classification from a management point of view. The preliminary topics for future works will be straightforward if attention paid to the duality of role and category abstractions. Following this duality, the model can be further extended to include concepts such as "category hierarchies" (as in [24]) and "separation of categories" as emulations of "role hierarchies" and "separation of duties". Automated enforcement of major access control policies is another area which is worth further studies. Especially, methods for combining different policies in systems adopting more than one major access control policy, such as a military hospital that must comply with both health-care and military standards. This can be a ground for combining policies designed from different points of view which is believed to be one of the limitations of current RBAC [13] and can open the way for a divide-and-conquer approach in policy design.

References

1. American National Standards Institute: American National Standard for Information Technology, Role Based Access Control, ANSI/INCITS 359 (2004)
2. Bell, D.E., Lapadula, L.J.: Secure Computer Systems: Mathematical Foundations, Mitre Corp., Bedford, MA, Technical Report ESD-TR-73-278 (1973)
3. Covington, M.J., Moyer, M.J., Ahamad, M.: Generalized Role-Based Access Control for Securing Future Applications. In: Proceedings of 23rd National Information Systems Security Conference, Baltimore, MD, October 2000 (2000)

4. Damiani, Ernesto, Vimercati, De Capitani Di, S., Paraboschi, Stefano, Samarati, Pierangela.: Design and Implementation of an Access Control Processor for XML Documents. In: Proceedings of the 9th International World Wide Web Conference on Computer Networks: the International Journal of Computer and Telecommunications Networking, pp. 59–75 (2000)
5. Damiani, Ernesto, Vimercati, De Capitani Di, S., Paraboschi, Stefano, Samarati, Pierangela.: A Fine-Grained Access Control System For XML Documents. ACM Transactions on Information and System Security 5(2), 169–202 (2002)
6. Ferraiolo, D.F., Kuhn, R.: Role-Based Access Control. In: Proceedings of the 15th NIST-NSA National Computer Security Conference, Baltimore, Maryland, October 1992, pp. 554–563 (1992)
7. Ferraiolo, D.F., Cugini, J.A., Kuhn, D.R.: Role-Based Access Control: Features and Motivations. In: Proceedings of the 11th Annual Computer Security Applications, New Orleans, LA, December 1995, pp. 241–248 (1995)
8. Ferraiolo, D.F., Barkley, J.F., Kuhn, D.R.: A Role-Based Access Control Model and Reference Implementation within a Corporate Intranet. ACM Transactions on Information and System Security 2(1), 34–64 (1999)
9. Ferraiolo, D.F., Sandhu, Ravi, Gavrila, Serban, Kuhn, D.R., Chandrmouli, Ramaswamy.: Proposed NIST Standard for Role-Based Access Control. ACM Transactions on Information and System Security 4(3), 224–274 (2001)
10. Ferraiolo, D.F., Kuhn, D.R., Chandramouli, Ramaswamy.: Role-Based Access Control, Artech House London (2003)
11. Giuri, Luigi, Iglio, Pietro.: Role Templates For Content-Based Access Control. In: Proceedings of the Second ACM Workshop on Role-Based Access Control, pp. 153–159 (1997)
12. Goh, Cheh, Baldwin, Adrian.: Towards a More Complete Model of Role. In: Proceedings of the Third ACM Workshop on Role-Based Access Control, pp. 55–62 (1998)
13. Hu, Ferraiolo, V.C., Kuhn, D.F., Rick, D.: Assessment of Access Control Systems, National Institute of Standard Technology, Interagency Report 7316 (2006)
14. International Standard Organization: Information Technology-Security Techniques-Code of Practice for Information Security Management, ISO/IEC 17799:2005 (2005)
15. International Standard Organization: Information Technology-Security Techniques-Information Security Management Systems Requirements, ISO/IEC 27001:2005 (2005)
16. Kumar, Arun, Karnik, Neeran, Chafle, Girish.: Context Sensitivity in Role-Based Access Control. ACM SIGOPS Operating Systems Review 36(3), 53–66 (2002)
17. Osborn, Sylvia, Sandhu, Ravi, Munawer, Qamar.: Configuring Role-Based Access Control to Enforce Mandatory and Discretionary Access Control Policies. ACM Transactions on Information and System Security 3(2), 85–106 (2000)
18. Roeckle, Haio, Schimpf, Gerhard, Weidinger, Rupert.: Process-Oriented Approach for Role-Finding to Implement Role-Based Security Administration in a Large Industrial Organization. In: Proceedings of the Fifth ACM Workshop on Role-based Access Control, pp. 103-110 (2000)
19. Sandhu, Ravi, Coyne, Edward. J., Feinstein, Hal, L., Youman, Charles, E.: Role-Based Access Control: A Multi-Dimensional View. In: Proceedings of 10th Annual Computer Security Applications Conference, December 1994, Orlando, Florida, pp. 54–62 (1994)
20. Sandhu, Ravi, Coynek, Edward, J., Feinsteink, Hal, L., Youmank, C.E.: Role-Based Access Control Models. IEEE Computer 29(2), 38–47 (1996)

21. Schaad, Andreas, Moffett, Jonathan, Jacob, Jeremy.: The Role-Based Access Control System of a European Bank: a Case Study and Discussion. In: Proceedings of the Sixth ACM Symposium on Access Control Models and Technologies, pp. 3–9 (2001)
22. Thomas, R.K.: Team-Based Access Control (TMAC): A Primitive for Applying Role-Based Access Controls in Collaborative Environments. In: Proceedings of the Second ACM Workshop on Role-Based Access Control, pp. 13–19 (1997)
23. Al-Kahtani, M.A., Sandhu, R.: Induced Role Hierarchies with Attribute-Based RBAC. In: Proceedings of the Eighth ACM Symposium on Access Control Models and Technologies, pp. 142–148 (2003)
24. Chae, J.: Towards Modal Logic Formalization of the Role-based Access Control with Object Classes. In: FORTE 2007. LNCS, vol. 4574, pp. 97–111. Springer, Heidelberg (2007)