

Models for Coalition-based Access Control (CBAC)

[Extended Abstract] *

Eve Cohen

NAI Labs at Network Associates, Inc.
3415 S. Sepulveda Boulevard, Suite 700
Los Angeles, California 90034

ecohen@nai.com

William Winsborough

NAI Labs at Network Associates, Inc.
3060 Washington Road
Glenwood, Maryland 21738

wwinsbor@nai.com

Roshan K. Thomas

NAI Labs at Network Associates, Inc.
1145 Herndon Parkway, Suite 500
Herndon, Virginia 20170

rthomas@nai.com

Deborah Shands

NAI Labs at Network Associates, Inc.
3415 S. Sepulveda Boulevard, Suite 700
Los Angeles, California 90034

dshands@nai.com

ABSTRACT

To effectively participate in modern coalitions, member organizations must be able to share specific data and functionality with coalition partners, while ensuring that their resources are safe from inappropriate access. This requires access control models, policies, and enforcement mechanisms for coalition resources. This paper describes a family of coalition-based access control (CBAC) models, developed to provide a range of expressivity with an accompanying range of implementation complexity. We define the protection state of a system, which provides the semantics of CBAC-based access policies. Finally, we briefly examine some of the issues for coalition access policy development and administration, and the complexity of implementing access enforcement mechanisms in a coalition environment.

Categories and Subject Descriptors

K.6.5 [Computing Milieux]: Management of Computing and Information Systems—*Security and Protection*; D.4.6 [Software]: Operating Systems—*Security and Protection*

General Terms

security, theory

Keywords

access control, security policy, domain models, coalitions, authorizations, roles, tasks, teams

*A full version of this paper is available as Technical Report #02-009, NAI Labs at Network Associates, March 2002.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SACMAT'02, June 3-4, 2002, Monterey, California, USA.
Copyright 2002 ACM 1-58113-496-7/02/0006 ...\$5.00.

1. INTRODUCTION

Businesses, governments, and other organizations are finding that coalition operations are increasingly critical to their success. Commercial coalitions may implement supply chain arrangements, subcontracting relationships, or joint marketing campaigns. International coalitions form to support joint defense objectives and peace-keeping efforts, as well as a variety of humanitarian, environmental, or trade-related goals. Such coalitions may be dynamic, as changing conditions and trust relationships result in new missions and modifications to coalition membership.

Because organizations rely on their information systems to support their operations, the ability to securely share information system resources has become critical to the tightly integrated systems and processes of business, military and other coalitions. Secure sharing requires that organizations be able to exercise fine-grained, policy-governed control over access to shared resources. Unfortunately, current technologies do not comprehensively support such control for coalition resource sharing. Though there is a variety of conceptual, technological and operational factors that contribute to this situation, we specifically address the following problem: current access control and domain models do not adequately capture inter-organization relationships.

This paper introduces the first collection of coalition-focused access control models. We present a family of access control models that capture the entities involved in coalition resource sharing, identify the interrelationships among those entities, and detail requirements for building authorizations in coalition environments. Such models are a necessary foundation for the development of coalition-focused access policies and enforcement mechanisms.

Our family of *coalition-based access control (CBAC)* models provides a broad spectrum of functionality, expressiveness and flexibility in support of coalition access control policies. Our basic CBAC model layers coalition access control concepts on top of a simple role-based access control (RBAC) model. The other CBAC models incorporate elements of team-based (TMAC) and task-based (TBAC) access control. These models support the use of system con-

text information in decisions to activate, synchronize and deactivate permissions. The CBAC family provides a suite of models which range in expressivity, as well as implementation complexity. A system architect may choose the most appropriate model for implementation or design a path for upgrading the security functionality of the system.

The remainder of this paper is structured as follows: Section 2 surveys related work, while Section 3 describes our coalition domain models and provides a working example. In Section 4, we sketch some of the formalisms defining the domain models and provide full definitions of the access control models. Section 5 discusses issues related to access control for coalition environments. Finally, Section 6 summarizes our results.

2. RELATED WORK

A variety of access control models have been developed over the years in response to system and security administration requirements. Some of the early access control models were developed to support security in operating systems. Thus, the early work described by Graham and Denning [10] and later refined by Harrison, Ruzzo and Ullman [11] introduced the notion of an access control matrix to represent conceptually the access rights of subjects to specific objects. Such models provide a system-centric, subject-object view of access control.

Subsequent research in access control models has been motivated by the need to reduce the security administration overhead commonly associated with low-level, subject-object permissions. Models that incorporate additional contextual information and support higher-level policy abstractions can simplify policy administration by reducing the semantic gap between enterprise-level policies and policies that can be directly enforced within a system. Abstractions such as “role,” “team,” and “task” were developed to model contextual information associated with organizational roles and responsibilities and collaborative activity.

Role-based access control (RBAC) [15] departed from subject-object models by introducing the organizational concept of “role,” abstracting a concept of a user’s job function within an organization. Users can then be assigned to roles, allowing an administrator to manage permissions on a few organizational roles rather than directly on a multitude of users.

Subsequent attempts to apply RBAC in collaborative environments revealed some of RBAC’s limitations. In particular, RBAC does not provide an abstraction to capture a set of collaborating users, operating in, potentially, different roles. The need for such an abstraction in collaborative environments led to the development of the team-based access control (TMAC) [18] model. In the TMAC model, access permissions can be specified on a “team” of collaborating users, acting in various roles. The notion of integrating the concept of a “team” with role-based access control was also pursued by Wang in [20], where the focus was cooperative hypermedia environments.

Access control models have also been developed to synchronize access permissions with ongoing tasks and workflow instances. As described in [19], the task-based authorization control (TBAC) model supports synchronization of permission activation and deactivation with changes in system state. This allows permissions to track the overall progress of a task and supports secure workflow management. A

workflow authorization model (WAM) presented in [1] has similar motivations. In WAM, an authorization links a permission to a time interval. The C-TMAC model presented in [8] has recently addressed the issue of incorporating broader contextual information (such as the location and time of access) into the TMAC model. The SALSA system [12] provides access control mechanisms for inter-organizational workflow to support the autonomy of the participating organizations. At design time, a single, cross-organizational workflow is created by the SALSA design tool. At runtime, however, that workflow is split into multiple, autonomous workflows, one for each participating organization. SALSA permits the grouping of related tasks into a more abstract, higher-level task, while access is controlled at the role level.

The CBAC models incorporate aspects of RBAC, TMAC and TBAC, along with coalition-targeted abstractions and context information. The additional abstractions support the expression of organization-level collaboration, beyond the user-level collaboration expressible in TMAC. These abstractions, along with appropriate constraints, also permit us to specify internal organizational structure (e.g., sub-organizations, sub-functions).

3. COALITION COMPONENTS

To build an effective model of access control for coalition sharing, we must first establish the coalition entities to be modeled and their relationships to one another. In selecting the entities and relationships, we form a model of the coalition domain of discourse. Section 4 then develops the CBAC access control models, built on our domain model.

In this section, we informally define the conceptual entities of our coalition domain model and discuss the intended relationships among those entities. The domain model is described in three views: the coalition, organization, and operations views. These views reflect the different perspectives of various coalition stake-holders on coalition activities and resources. A sketch of the formalisms defining the domain model appears in Section 4, while the complete formal definitions appear in [3]. To unify our informal and formal presentations of the model domain, `DOMAINENTITIES` will be shown in a small capitals font in the text below.

3.1 Coalition-level Domain Entities

Coalition entities are defined at the executive level where international diplomacy or top-level corporate agreements and their associated abstractions are specified.

A `COALITION` is a collection of partner organizations associated voluntarily, and often temporarily, to accomplish one or more missions. A `PARTNERORGANIZATION` is an autonomous entity that is free to make agreements on its own behalf (e.g., a nation, a corporation). A `MISSION` is a goal that may be very broad or very specific. `PRINCIPALFUNCTIONS` are the fundamental classes of activities required to accomplish the `MISSION`. `PARTNERORGANIZATIONS` commit various `ORGANIZATIONASSETS` to the accomplishment of those `PRINCIPALFUNCTIONS`. Subsets of the larger coalition membership may form bi-lateral or multi-lateral communities of interest (COI), based on common interests, or established levels of trust. Resource sharing among organizations within a community of interest may be more extensive than within the full coalition.

Figure 1 shows the coalition view of the domain model. Coalition entities appear as boxes in the figure, while rela-

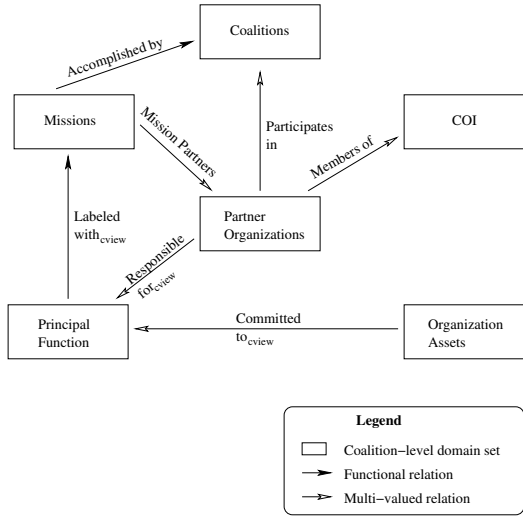


Figure 1: Domain Model, Coalition View

tions between those entities are shown with arrows. Solid-headed arrows indicate functional relations, while arrows with open-heads indicate multi-valued relations. Section 4 provides informal and formal definitions of some of the relations.

3.2 Organization-level Domain Entities

The organization view reflects the structures supporting coalition activities, as seen by management personnel within a partner organization. Organizational-level entities are defined at various levels within an organization (e.g., business unit, department, division). An organization entity might refine a coalition view entity or introduce a concept necessary to the implementation of coalition agreements.

The PARTNERORGANIZATIONS, identified within the coalition view, may be subdivided into an arbitrary tree structure of ORGANIZATIONS, permitting organizational hierarchies (e.g., a group within a department within a division) to be accurately modeled. Similarly, PRINCIPALFUNCTIONS may be further segmented into a partially ordered structure of FUNCTIONS. ORGANIZATIONASSETS may be segmented into a partially ordered structure of ORGANIZATIONRESOURCES. The ability to specify these hierarchical relationships supports the definition of a consistent hierarchy of views within an Organization: the group manager’s view shows resources that are within (i.e., components of) the resources that appear in the department manager’s view.

In the organization view, three new types of entities are introduced: ROLES, TEAMS and TASKS. A ROLE captures a coherent aspect of an individual’s job function within the organization. As defined in [18], a TEAM is a collection of users assigned to various roles and working toward the accomplishment of a specific goal. The team definition specifies the roles that will be included within the team. A TASK, defined in [19], is a stateful flow of activities that achieve a particular function. A TASK may address multiple FUNCTIONS and a FUNCTION may be achieved through multiple TASKS.

Figure 2 shows the organization-level view of the domain

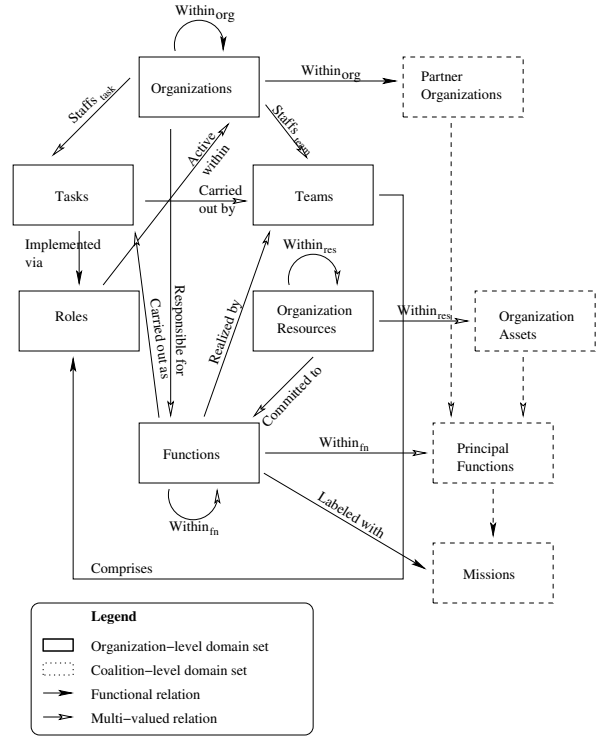


Figure 2: Domain Model, Organization View

model. Note that portions of Figure 1 are shown again (with dashed lines) in Figure 2 to permit a convenient representation of relations that span the two views.

3.3 Operations-level Domain Entities

The operations view is concerned with the relationships between users and organization entities. Most security administration of computer systems is likely to take place at this level.

Only a single new coalition entity is introduced to the coalition domain model in the operations view: the USER. Users are employed by organizations, perform in roles, are employed on functions, and are, potentially, assigned to teams or to contribute to tasks.

Figure 3 shows the operations-level view of the domain model. As before, portions of Figure 2 appear again (with dashed lines) in Figure 3 to permit a convenient representation of relations between USERS and the organization-level entities.

3.4 Domain Model Example

In this section, we illustrate the use of the coalition domain model and its three views with descriptions of the coalition activities and structure of the U. S. Federal Emergency Management Agency (FEMA)[6]. FEMA’s strategic plan [4] is defined in terms of the coalitions, both standing and *ad hoc*, in which it participates. In a wide variety of different contexts, FEMA cooperates with and coordinates the activities of federal, state and local agencies together with the American Red Cross and other non-governmental organizations.

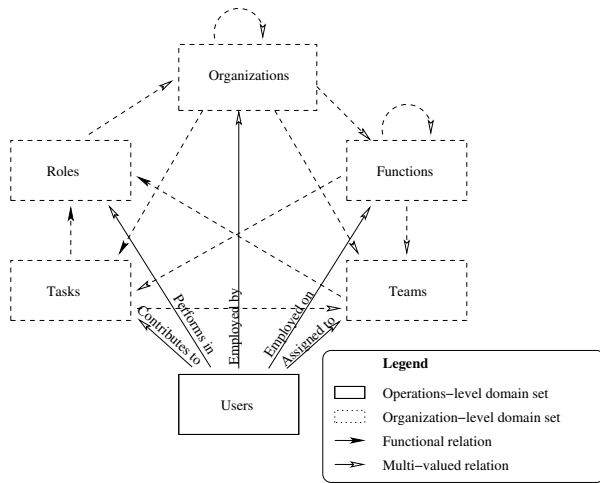


Figure 3: Domain Model, Operations View

3.4.1 FEMA’s Coalition-level Domain Entities

In our domain model, coalition-level entities include COALITIONS, PARTNERORGANIZATIONS, MISSIONS, PRINCIPALFUNCTIONS, ORGANIZATIONASSETS, and COI. The following briefly describes a few coalitions that FEMA coordinates, their associated missions, and the principal organizations that participate in them.

- The Federal Response Plan (FRP)[9] COALITION with FEMA is a collection of 28 PARTNERORGANIZATIONS, all governmental agencies led by FEMA. This COALITION of agencies has as its MISSION the maintenance of the FRP. The FRP encompasses the system for cooperation among the PARTNERORGANIZATIONS in response and recovery phases of disasters.
- The National Earthquake Hazards Reduction Program (NEHRP) COALITION is a multi-agency federal effort the MISSION of which is to reduce the nation’s losses due to earthquakes. The principal program agencies cooperating as PARTNERORGANIZATIONS in this COALITION are the U.S. Geological Survey, the National Science Foundation, and the National Institute of Standards and Technology; they are joined by many contributing departments from other governmental agencies.
- Through Project Impact—Disaster Resistant Communities [7], FEMA has led and supported efforts by communities to become disaster resistant. These COALITIONS of local governmental and private groups have combined in a variety of ways to make communities and local businesses less subject to natural disasters.

Below, we focus on coalitions that form under Project Impact and consider some of the organization assets and principal functions of those coalitions. Started as a pilot program in seven communities in 1997, Project Impact has become a nationwide initiative including about 250 community coalitions. Each of these coalitions includes local government and business interests which partner with FEMA and other

federal and state governmental departments in support of the Project Impact goals.

Among the ORGANIZATIONASSETS that FEMA commits to a COALITION are information on how to conduct the coalition’s MISSION and funding to assist in accomplishing the MISSION.

The three PRINCIPALFUNCTIONS for any Project Impact COALITION are

- Hazard and Risk Assessment,
- Disaster Planning, and
- Mitigation.

Within Project Impact, COIs form primarily based on location. For example, the Miami-Dade Working Group (for Miami-Dade county, Florida) is made up of 31 municipalities, county departments, businesses, colleges and universities, not-for-profit organizations and state and federal agencies.

3.4.2 FEMA’s Organization-level Domain Entities

In our domain model, organization-level entities include ORGANIZATIONS, FUNCTIONS, ORGANIZATIONRESOURCES, ROLES, TEAMS, and TASKS. We now look inside of FEMA to see examples of these entities.

FEMA is structured into regional offices. The Region I Office, for example, is located in Boston and serves New England [5]. The Region I Office is an ORGANIZATION within the FEMA PARTNERORGANIZATION, with responsibilities to the Project Impact COALITION. Within the Region I Office are a number of (sub) organizations, including the Mitigation Branch.

The Mitigation Branch works on a number of FUNCTIONS subordinate to the PRINCIPALFUNCTION Mitigation. In particular, the Mitigation Branch works on the following FUNCTIONS:

- Flood Mitigation,
- Fire Mitigation,
- Hazardous Materials Mitigation, and
- Earthquake Damage Mitigation.

The Mitigation Branch staffs the standing Community Mitigation Team. Recent TASKS undertaken by this TEAM in support of Flood Mitigation are the assessment, design and construction of the Quincy, Massachusetts Hollis Avenue pumping station and of a flood wall in North Reading, Massachusetts.

3.4.3 FEMA’s Operations-level Domain Entities

In our domain model, the Operations view focuses on the relationships between USERS and organization-level entities. We now consider the users that are part of the Mitigation Branch’s Community Mitigation Team.

The TEAM members, listed here with some of their ROLES, are USERS within the Mitigation Branch ORGANIZATION:

- Paul Ford, Hazard Mitigation Specialist
Regional National Flood Insurance Coordinator
Insurance Liaison

- Jim Gibbons, Hazard Mitigation Specialist
Community Rating System Administrator
Community Assistance Program Liaison
- George Hatch, Hazard Mitigation Specialist
Community Compliance Specialist
NFIP Liaison

The relationships among FEMA’s coalition-, organization- and operations-level entities can generally be expressed using the relations and functions of the CBAC coalition domain model. For example, the Mitigation Branch ORGANIZATION staffs the Community Mitigation TEAM and each of the above USERS performs in specific ROLES. Overall, the CBAC coalition domain model appears to support a natural expression of FEMA’s coalition-focused structure.

4. THE CBAC MODELS

Our approach to constructing a family of access control models is based on the comprehensive coalition domain model described in Section 3. Distinctions between the domain models focus on the inclusion (or exclusion) of the TEAMS and TASKS entities within the domain model. The basic model includes neither. Two additional models include either TEAMS or TASKS, but not both. Finally, the most complex of the models includes both TEAMS and TASKS. These four coalition domain models underlie the CBAC family of access control models, named: $CBAC_{basic}$, $CBAC_{team}$, $CBAC_{task}$, and $CBAC_{team+task}$.

In this section, we sketch some of the formalisms defining the CBAC domain models. An instance of any one of the domain models gives a collection of sets, where each set corresponds to one of the boxes in Figures 3.1, 2, or 3. The instance also gives several relations, each of which corresponds to one of the arrows in the figures. The domain model imposes several constraints on these relations, which serve to ensure that the relationships between the domain entities are coherent and meaningful. Space constraints limit our treatment of the domain models, however their complete formal definitions can be found in [3]. We begin by defining the domain model for each member of the CBAC family and identifying a few illustrative relationships among the domain entities, as well as some of the constraints on those relationships. After describing each of the four domain models, we develop the access control models. In particular, we introduce the concepts of “authorization set” and “protection state,” and identify necessary constraints.

In the model definitions below, we make use of some additional notational conventions. **Relations** among domain entities are represented below in a bold font and **parameters** and bound variables are shown below in a sans serif font. $\mathcal{P}(X)$ is used to denote the power set of a set X , that is to say, the set having as its members all subsets of X . A dot “.” is a scoping notation and stands for a left bracket whose mate is as far to the right as is possible without altering the pairing of left and right brackets already present.

4.1 The $CBAC_{basic}$ Domain Model

$CBAC_{basic}$ is built on the simplest domain model in the CBAC family, adding coalition entities to a simple RBAC model. It includes the domain entities COALITIONS, PARTNERORGANIZATIONS, MISSIONS, PRINCIPALFUNCTIONS, ORGANIZATIONASSETS, COI, ORGANIZATIONS, FUNCTIONS,

ORGANIZATIONRESOURCES, ROLES and USERS and relations among them. For example:

- **ParticipatesIn**: $PARTNERORGANIZATIONS \times COALITIONS$ holds for (p, c) when p is a member of c .

Example: The US Army has signed a memorandum of agreement with FEMA to participate in Project Impact. Thus,

$$(Army, Project Impact) \in \mathbf{ParticipatesIn}$$

- **AccomplishedBy**: $MISSIONS \rightarrow COALITIONS$ maps each mission to the coalition that has accepted it.

Example: One of the specific MISSIONS recently undertaken by Project Impact is the repair of the Pajaro River levee in northern California. Thus,

$$\mathbf{AccomplishedBy}(\text{Pajaro River Flood Mitigation}) = \text{Project Impact}$$

- **MissionPartner**: $MISSIONS \times PARTNERORGANIZATIONS$, holds for (m, p) when p is involved in the execution of m . The consistency of **ParticipatesIn**, **AccomplishedBy** and **MissionPartner** is assured by the following constraint, which states that any partner in a mission participates in the coalition which is to accomplish that mission:

$$\forall m \in MISSIONS \forall p \in PARTNERORGANIZATIONS .$$

$$\mathbf{MissionPartner}(m, p) \Rightarrow$$

$$\mathbf{ParticipatesIn}(p, \mathbf{AccomplishedBy}(m))$$

Example: This constraint ensures that when we say that the Army is a partner in the Pajaro River Flood Mitigation mission, then we can be sure that the Army is participating in Project Impact, which is accomplishing the Pajaro River Flood Mitigation mission.

By grouping together users acting in roles, involved in coalition activities, $CBAC_{basic}$ provides important links between the individual users of the operations view and the organizations of the organization and coalition views. However, this relatively simple framework for modeling relationships among users and their organizations offers very little support for identifying relationships between collections of users or their activities. To better support such relationships, we developed the $CBAC_{team}$, $CBAC_{task}$, and $CBAC_{team+task}$ models, presented below.

4.2 The $CBAC_{team}$ Domain Model

The $CBAC_{team}$ adds the domain entity TEAMS to the domain entities specified in the $CBAC_{basic}$ model and, of course, adds additional relations to integrate TEAMS with the other domain entities. The set of relations of $CBAC_{team}$ includes the relations of $CBAC_{basic}$, in addition to those that relate members of the TEAMS domain to other domain entities. For example:

- **Staffs_{team}**: $ORGANIZATIONS \times TEAMS$ holds for (org, tm) when org shares the responsibility for staffing tm .

Example: As a sub-organization of a Project Impact partner, the Army Corps of Engineers provides personnel to support projects like the Pajaro River levee repair. Thus,

$$(Army Corps of Engineers, Pajaro River levee team) \in \mathbf{Staffs}_{team}.$$

- For any function, team and organization, if the function is partially accomplished by the team and the organization has staffing responsibility for the team, then that organization must also be responsible for the function. Such constraints ensure that the TEAMS abstraction is used in a way that maintains critical links between ORGANIZATIONS and FUNCTIONS:

$$\begin{aligned} & \forall fn \in \text{FUNCTIONS} \forall tm \in \text{TEAMS} \\ & \forall org \in \text{ORGANIZATIONS} . \\ & \quad \mathbf{RealizedBy}(fn, tm) \wedge \mathbf{Staffs}_{team}(org, tm) \Rightarrow \\ & \quad \quad \mathbf{ResponsibleFor}(org, fn) \end{aligned}$$

Example: This constraint ensures that if the Army Corps of Engineers is not at least partially responsible for a levee repair function, then either the Corps does not provide staff to the Pajaro River levee team or the Pajaro River levee team is not working on the levee repair function.

The introduction of the TEAMS domain enables the expression of cross-organizational relationships among users acting in roles. The “team” concept embodies a more collaborative model of coalition activity than that of a collection of independent organizations, each of which may work independently on some common functions. Coalitions in which fine-grained collaboration among participants is common are more likely to require the “team” concept.

4.3 The CBAC_{task} Domain Model

CBAC_{task} adds the domain entity TASKS to the domain entities specified in the CBAC_{basic} model and, of course, adds additional relations to integrate TASKS with the other domain entities. The set of relations of CBAC_{task} includes the relations of CBAC_{basic}, in addition to those that relate members of the TASKS domain to other domain entities. For example, **ImplementedVia**: TASKS \rightarrow \mathcal{P} (ROLES) maps **ta** \in TASKS to the set **rset** \subseteq ROLES needed in the performance of **ta**. The following restriction holds for **ImplementedVia**: if a task is implemented via a set of roles, then each of the roles must be active within some organization with staffing responsibility for the task.

$$\begin{aligned} & \forall ta \in \text{TASKS} \forall rset \in \mathcal{P}(\text{ROLES}) . \\ & \quad \mathbf{ImplementedVia}(ta) = rset \Rightarrow \\ & \quad \forall r \in rset \exists org \in \text{ORGANIZATIONS} . \\ & \quad \quad \mathbf{ActiveWithin}(r, org) \wedge \mathbf{Staffs}_{task}(org, ta) \end{aligned}$$

Example: For a task such as Flood Risk Assessment, a number of specialists are required. Suppose,

$$\begin{aligned} & \mathbf{ImplementedVia}(\text{Flood Risk Assessment}) = \\ & \quad \{\text{Hydrologist, Engineer, Hazard Mitigation Specialist}\} \end{aligned}$$

The constraint says that, for example, there must be at least one organization staffing the Flood Risk Assessment task in which the role Hydrologist is active.

The addition of the TASKS domain introduces a notion of system state, as an instance of a task may be in any one of a number of states (depending on the system context) and multiple tasks may be necessary to accomplish a function. As detailed in [19], permissions may then be activated and deactivated, depending on task state.

4.4 The CBAC_{team+task} Domain Model

CBAC_{team+task} is the most comprehensive of the CBAC models, including all of the domain entities of CBAC_{basic}, in addition to the TEAMS and TASKS domains. The set of relations of CBAC_{team+task} includes the relations of CBAC_{team}

and CBAC_{task}, in addition to **CarriedOutBy**. Constraints are also added to restrict the interaction of tasks and teams. For example:

CarriedOutBy: TASKS \times TEAMS holds for (**ta**, **tm**) when **tm** is assigned to work on **ta**. **CarriedOutBy** refines both **Staffs_{team}** and **Staffs_{task}**, in that if a task is to be carried out by a team, then an organization has staffing responsibility for the task if and only if it also has staffing responsibility for the team.

$$\forall ta \in \text{TASKS} \forall tm \in \text{TEAMS} . \mathbf{CarriedOutBy}(ta, tm) \Rightarrow \forall org \in \text{ORGANIZATIONS} .$$

$$\mathbf{Staffs}_{team}(org, tm) \Leftrightarrow \mathbf{Staffs}_{task}(org, ta)$$

If a user contributes to a task, he must do so as a member of a team:

$$\begin{aligned} & \forall ta \in \text{TASKS} \forall u \in \text{USERS} . \mathbf{ContributesTo}(u, ta) \Rightarrow \\ & \quad \exists tm \in \text{TEAMS} . \mathbf{CarriedOutBy}(ta, tm) \\ & \quad \wedge \mathbf{AssignedTo}(u, tm) \end{aligned}$$

Example: When FEMA hydrologist, Dave Knowles contributes to the Flood Risk Assessment task, he does so as part of the Pajaro River Levee Team to which he has been assigned.

By supporting the specification of relationships between teams and tasks, CBAC_{team+task} allows the activation of team permissions to be tied to task state. Thus, team members may be authorized to access a given resource only when one of the tasks that uses the resource is active.

4.5 CBAC Access Control Models

In the previous sections, we have discussed components of the coalition domain models: conceptual elements designed to reflect our understanding of “real world” ingredients of coalitions. In this section, we discuss access control models, built on top of the domain models by adding conceptual elements designed to support a semantics for coalition access policies. These elements provide the formal “overhead” necessary for precisely expressing “who” should have access to “what” and under “which circumstances.”

4.5.1 Authorization Sets

Informally, an *authorization* for a given resource **r** specifies a strictly positive association between “subject material” (including a name or identifier for a coalition, partner organization, mission, etc., along with the usual user identifier) and a permission that is appropriate to **r**. Though we do not formally interpret the PERMISSIONS for **r** \in ORGANIZATION-RESOURCES, PERMISSIONS[**r**] corresponds intuitively to the set of operations which might be applied to **r** (e.g., for a file system resource, PERMISSIONS[**r**] might include “read,” “write,” and “execute.”) An *authorization set* for a given resource is the collection of all such associations, across all states of the system and environment.

Formally, let us begin by defining for each domain model an *authorization space*, the cross product of the relevant domain entities:

$$\begin{aligned} \text{AUTHSPACE}_{basic}[r] = & \text{COALITIONS} \times \text{PARTNERORGANIZATIONS} \times \text{MISSIONS} \\ & \times \text{PRINCIPALFUNCTIONS} \times \text{ORGANIZATIONASSETS} \\ & \times \text{COI} \times \text{ORGANIZATIONS} \times \text{FUNCTIONS} \\ & \times \text{ORGANIZATIONRESOURCES} \times \text{ROLES} \times \text{USERS} \\ & \times \text{PERMISSIONS}[r] \end{aligned}$$

$$\text{AUTHSPACE}_{team}[r] = \text{AUTHSPACE}_{basic}[r] \times \text{TEAMS}$$

$$\text{AUTHSPACE}_{task}[r] = \text{AUTHSPACE}_{basic}[r] \times \text{TASKS}$$

$$\text{AUTHSPACE}_{team+task}[r] = \text{AUTHSPACE}_{team}[r] \times \text{TASKS}$$

For each domain model, its **AuthSet**[**r**] relation will be a

subset of the associated $\text{AUTHSPACE}[r]$. Each tuple $d = \langle d_1, d_2, \dots, d_n \rangle \in \text{AuthSet}[r]$ we call an *authorization*.

4.5.2 Constraints on the Authorization Set

Of course, an arbitrary subset of one of the domain cross products shown above would not necessarily represent a sensible set of authorizations under one of the CBAC models. In this section, we introduce constraints on authorization sets to restrict them to sets consistent with our intended semantics of the CBAC domain models.

Since an authorization set is a relation whose domain is a specific resource, we assert Constraint 1 to ensure that the authorizations in $\text{AuthSet}[r]$ pertain uniquely to resource r .

- **Constraint 1** (Consistent References to Resources): For any authorization $d \in \text{AuthSet}[r]$, d 's value in the `ORGANIZATIONRESOURCES` domain must be r .

Authorization sets should respect the relations between domain entities defined earlier in Section 4 and the constraints on those relations. Constraint 2 ensures that authorizations appearing in $\text{AuthSet}[r]$ satisfy those relations.

- **Constraint 2** (Coalition Relation Enforcement): Suppose that $D \subseteq D_1 \times D_2 \times \dots \times D_n$ is the domain for $\text{AuthSet}[r]$, an authorization set for model $M \in \{\text{CBAC}_{\text{basic}}, \text{CBAC}_{\text{team}}, \text{CBAC}_{\text{task}}, \text{CBAC}_{\text{team+task}}\}$. Then for every relation $\mathbf{R}: D_i \times D_j$ defined for M , $d = \langle d_1, d_2, \dots, d_n \rangle \in \text{AuthSet}[r]$ implies $\mathbf{R}(d_i, d_j)$.

Example: A database of building materials specifications could be one of the `ORGANIZATIONRESOURCES` managed by FEMA under `Project Impact`. An authorization for the database, which would be represented formally as an element of

$\text{AuthSet}_{\text{basic}}[\text{Materials Specification Database}]$, might look like the following:

(Project Impact, Army, Pajaro River Flood Mitigation, ..., Sgt. Alice Parker, read)

To meet Constraint 2, we must examine the authorization tuple to check that it respects all of the relations defined between domain entities. For example,

(Army, Project Impact) \in **ParticipatesIn**

and

AccomplishedBy(Pajaro River Flood Mitigation)
= Project Impact

Constraints 1 and 2 prevent authorizations that do not conform to our coalition domain models from being included in an authorization set.

4.5.3 Contexts and the Protection State

In this section, we complete the task of defining elements of the CBAC access control models necessary to provide semantics for access policies. We introduce notions of context and protection state to identify the set of authorizations that are in effect within a specific context.

Context is intended to represent the state of the system and its environment, including the user session (as typically defined in RBAC models) and the activation status of coalitions, missions, functions, roles and teams. It may also contain information such as location or time. The operational definition of context is heavily dependent on system implementation, so we will not specify a notation or semantics for it. Instead, for each resource r , we posit the existence

of a function **ContextFilter** $[r]$ that takes a context and set of authorizations and returns the subset of those authorizations that are valid in the given context.

ContextFilter $[r]$:

$\text{CONTEXT} \times \mathcal{P}(\text{AUTHSPACE}[r]) \rightarrow \mathcal{P}(\text{AUTHSPACE}[r])$

We can now use the **ContextFilter** $[r]$ function to define the *protection state* of resource r within a specific system context to be the set of authorizations that are valid in that context:

$\text{PS}[r](\text{context}) \equiv \text{ContextFilter}[r](\text{context}, \text{AuthSet}[r])$

The protection state of a specified resource precisely identifies the authorizations that are valid within the given context: if an authorization is missing from the protection state, then such access is not permitted. An information system based on any one of the CBAC models must implement the semantics defined by the protection state. Similarly, a language for expressing CBAC access policies must have its semantics based on the protection state.

The development of an appropriate CBAC-based policy expression language and runtime environment remains a task for future work. The structure of the domain model anticipates that abstract policies will be described in the vocabulary of the coalition view, later to be refined—that is, further restricted—at successively lower levels. For example, an organization such as FEMA may commit an organization asset of `Structural Guidelines` to a `Structural Solutions` function as part of the `Pajaro River Flood Mitigation` mission. This would not mean, however, that each coalition worker would have unlimited read and write access to each of the guidelines. Rather, the pertinent authorization sets would be refined through additional policy definition at the organization and operations level by sub-function, role, team or task membership and/or user identification. The policy expression language, runtime system, and policy authors must ensure that statements of policy result in appropriate additions to or deletions from the authorization set of a resource. Thus, high-level policy expressions may add many authorizations to the authorization set of a resource, while lower-level policy expressions effectively remove some of those authorizations.

To evaluate access policies based on one of the CBAC models, a system must implement the **ContextFilter** function to extract information critical to access decisions. The approach to such implementations may vary greatly from system to system. For example, one implementation may use the IP address of a request to identify the requestor's organization name, while another implementation may extract the organization name from an X.509 identity certificate. The sophistication of different **ContextFilter** implementations may also vary greatly. For example, one system may track the detailed progress of a mission or function through the firing of events in an active database, while another system relies on an administrator to manually identify the start and finish of these activities. By allowing various implementations of the **ContextFilter** function, we allow the CBAC models to be applied to a wide range of systems.

5. DISCUSSION

In the process of developing the CBAC family, we mapped existing descriptions of business, government, and international coalitions and collaborative structures (e.g., Covisint,

FEMA, NATO) onto our draft models. Often, these exercises helped us to improve our models by highlighting relations among coalition entities or constraints on those relations that we had previously overlooked. In acting the part of a CBAC “administrator,” we also identified a number of issues for CBAC and, more generally, for access control in coalition environments. In this section, we identify a few of those issues and discuss their impact, whether conceptual or in implementations.

5.1 CBAC Policy and Administration

The CBAC family of models offers a selection of paradigms for coalition-focused access control. To express access policies that have CBAC-based semantics, we are in the process of developing a policy expression language. To write access policies that are acceptable to coalition participants, we also need an administrative model for CBAC. An administrative model specifies how policy is developed and maintained. In particular, it answers questions regarding which entities are authorized to specify policy for which resources. See [14], for example, in which an administrative model was developed for RBAC. Within an organization that commonly practices delegation of authority along the organizational hierarchy, an administrative model might assume that a high-level administrator (i.e., an administrator representing the highest level of the organization) has broad authority to specify access policy for all resources “owned” by the organization and may delegate authority for some policy specification to lower-level administrators. Such a hierarchical model does not extend well to a coalition environment, as coalitions do not include a top-level organization in which ultimate authority is vested. In coalitions, administrative authority must be somewhat distributed among the partner organizations and that distribution may be negotiated. Partner organizations may use more hierarchical administrative models internally.

An administrative model that supports both hierarchical and distributed delegation of authority will not be trivial to develop. Consider, for example, an organization A that participates in coalition C. A-users have authorizations to coalition resources and A-administrators define roles and assign users in roles to teams that operate within the coalition. In defining roles and assigning users to those roles, A-administrators delegate authority within their own organization. In authorizing access to a resource by coalition teams (some of whose members may be employed by other organizations), A-administrators are delegating some authority over those authorizations to the organizations employing the team members. If A is ejected from the coalition, it is probably necessary to remove all authorizations held by A-users from authorization sets for any of the coalition resources. Must role-based or team-based authorizations be eliminated as well? If so, which entities are authorized to perform this policy change? The development of acceptable answers to such complex administrative questions is critical to the effectiveness of information resource sharing within coalitions. Such issues will require significant future research.

5.2 Implementation Complexity

The implementation of some CBAC models by access policy enforcement mechanisms may prove challenging. In particular, system context information, including user sessions and the activation states of missions, functions, tasks, etc.,

may be complex to implement. The family of CBAC models was developed to enable tradeoffs between expressive power and implementation complexity. We expect that, in the simplest case, a context-free variant of CBAC_{basic} could be implemented with defined missions, functions, teams, and roles in a perpetually active state, eliminating the need to manage activation states. On the other hand, the implementation of models supporting teams or tasks poses a variety of implementation challenges. We group these challenges into three categories: those that arise from the distributed computing environment, those that arise from active management of authorizations, and those that arise from the need for organizational autonomy within coalitions.

The first category of implementation challenges results from the complexity of distributed computing. For example, events that trigger changes in task and team state may originate from information systems in different partner organizations. Detecting and correlating those events to identify state changes is necessary to monitor task progress. The often tightly coupled events of workflow systems exacerbate the problems of identifying and maintaining a consistent view of task state within a distributed system. [12] discusses many of the issues involved with integrating access control into a distributed, inter-organizational workflow environment.

The second category of implementation challenges originates from the need to support active management of authorizations. For example, both the CBAC_{team} and CBAC_{task} models call for state-based activation and deactivation of permissions in accordance with the current state of tasks and teams, respectively. A delay (or failure) in the granting of an authorization may result in the rescheduling of certain workflow tasks. Conversely, the failure of a task or its sub-tasks may result in certain permissions not being activated and granted. Thus, the liveness and termination properties of distributed state management systems may impact the robustness and safety of active authorization management.

The third category of implementation challenges results from the need to support organizational autonomy within coalitions. A variety of challenges fall into this category, among them, the need for trust management and distributed delegation of authority. Collaborating, autonomous organizations are likely to want and need to cede some authority to one another. For instance, one organization may need to trust another concerning which employees it assigns to various roles. Systems to support delegation of authority and specification and evaluation of trust policies [2, 13], as well as systems to distribute and enforce coalition access control policies [16, 17] will be needed.

Overall, a complete implementation of the CBAC access control models will involve the integration of security mechanisms with a wide variety of platforms, network infrastructures, workflow applications, and distributed computing systems across organizational boundaries. Much of the complexity involved in implementing CBAC models arises from the inherent complexity of coalition-based interactions. Implementations of other access control models face the same complex issues, however, when they are used within distributed coalition environments.

Consider, for example, a coalition that uses an RBAC access control model to coordinate members’ activities toward joint objectives. Roles, constraints, hierarchies, etc. must be used to encode any necessary coordination infor-

mation. Thus role names such as Pajaro River hydrologist or Pajaro River engineer may be used to indicate a team-like relationship. Of course, such roles must be activated at the appropriate times (e.g., simultaneously, or in a specific sequence) to support a collaborative application. Thus, role activation must be managed according to the state of the distributed system. In a coalition using RBAC, the requirement for such distributed state-based activation management is implicit, falling to the distributed system infrastructure.

In general, access control for coalition environments places complex, **implicit** requirements on implementations. By directly codifying essential aspects of coalition-focused access control (e.g., organizational autonomy, negotiated agreements, shared responsibilities), the CBAC models capture **explicit** implementation requirements. This may improve the chances that implementations will meet the requirements.

6. CONCLUSION

This paper has introduced the CBAC family of access control models to capture the semantics of coalition interrelationships and the characteristics of access control in such environments. CBAC_{basic} is the simplest model in the family, adding coalition entities and relationships to a role-based model. CBAC_{team} builds on CBAC_{basic} to support the collection of users, acting in roles, into teams. CBAC_{task} builds on CBAC_{basic} to allow access to be determined based on task state. Finally, CBAC_{team+task} combines the concepts of both teams and tasks to enable team activation based on task state. This family of models provides a spectrum of expressiveness, with an accompanying spectrum of implementation complexity.

The CBAC models are based on a coalition domain model that identifies coalition entities and their interrelationships. Relationships between coalition entities are constrained, according to our understanding of what is (and is not) feasible in the “real world.” The description of our domain model provides a formal expression of our assumptions about coalition operations and offers a basis for evaluating the applicability of our access control models to a given coalition environment.

We have defined the notion of a protection state to capture authorizations that are valid within a given context. Authorization sets are constrained to include only those authorizations that match our domain models. The protection state provides semantics for CBAC-based access policies—any system that enforces CBAC-based policies must implement this protection state.

Finally, we have identified future work necessary to support deployment of CBAC-based systems, as well as broader issues affecting access control within coalition environments.

7. ACKNOWLEDGEMENTS

We wish to thank the referees for catching a notational error in our domain model and providing other helpful comments. This work was supported by DARPA through the Space and Naval Warfare Systems Center – San Diego under Contract No. N66001-00-C-8070.

8. REFERENCES

- [1] V. Atluri and W.-K. Huang. An authorization model for workflows. In E. Bertino, H. Kurth, G. Martella, and E. Montolivo, editors, In *Proceedings of the Fourth European Symposium on Research in Computer Security*, LNCS, pages 44–64, Rome, Italy, September 1996.
- [2] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. D. Keromytis. The KeyNote trust-management system, Version 2. Request for Comments 2704, Internet Engineering Task Force, September, 1999.
- [3] E. Cohen, R. Thomas, W. Winsborough, and D. Shands. Models for coalition-based access control (CBAC). Technical Report #02-009, NAI Labs at Network Associates, March 2002.
- [4] FEMA. Fema strategic plan FY 2000—FY 2006. In http://www.fema.gov/library/splan_01.htm/, 2000.
- [5] FEMA. FEMA Region I Website. In <http://www.fema.gov/Reg-I/index.htm>, 2001.
- [6] FEMA. FEMA website. In <http://www.fema.gov/>, 2001.
- [7] FEMA. Project Impact. In <http://www.fema.gov/impact/>, 2001.
- [8] C. Georgiadis, I. Mavridis, G. Pangalos, and R. K. Thomas. Flexible team-based access control using contexts. In *Proceedings of the Sixth ACM Symposium on Access Control Models and Technologies*, pages 21–27, Chantilly, Virginia, May 2001.
- [9] U. Government. Federal response plan. In <http://www.fema.gov/r-n-r/frp/>, April 1999.
- [10] G. Graham and P. Denning. Protection—principles and practice. In *Proceedings of the AFIPS Spring Joint Computer Conference*, volume 40, pages 417–429, Atlantic City, New Jersey, May 1972.
- [11] M. H. Harrison, W. L. Ruzzo, and J. D. Ullman. Protection in operating systems. *Communications of the ACM*, 19(8):461–471, 1976.
- [12] M. H. Kang, J. S. Park and J. N. Froscher. Access control mechanisms for inter-organizational workflow. In *Proceedings of the 6th ACM Symposium on Access Control Models and Technologies*, pages 66–74, Chantilly, Virginia, May 2001.
- [13] N. Li, J. C. Mitchell, and W. H. Winsborough. Design of a role-based trust management framework. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, Oakland, California, May, 2002.
- [14] R. Sandhu, V. Bhamidipati, and Q. Munawer. The ARBAC97 model for role-based administration of roles. *ACM Transactions on Information System Security*, 2(1):105–135, 1999.
- [15] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, February 1996.
- [16] D. Shands, R. Yee, J. Jacobs, and E. J. Sebes. Secure virtual enclaves: Supporting coalition use of distributed application technologies. In *Proceedings of the Network and Distributed Systems Security Symposium (NDSS 2000)*, pages 187-202, San Diego, California, February, 2000.

- [17] D. F. Sterne, G. W. Tally, C. D. McDonell, D. L. Sherman, D. L. Sames, and P. X. Pasturel. Scalable access control for distributed object systems. In *Proceedings of the Eighth USENIX Security Symposium*, Washington, D. C., August, 1999.
- [18] R. K. Thomas. 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*, pages 13–19, Fairfax, Virginia, November 1997.
- [19] R. K. Thomas and R. Sandhu. Task-based authorization controls (TBAC): A family of models for active and enterprise-oriented authorization management. In *Proceedings of the IFIP WG 11.3 Workshop on Database Security*, pages 166–181, Lake Tahoe, California, August 1997.
- [20] W. Wang. Team- and role-based organizational context and access control for cooperative hypermedia environments. In *Proceedings of ACM Hypertext '99*, pages 37–46, Darmstadt, Germany, February 1999.