Delegation in Role Based Access Control Model for Workflow Systems

Prasanna H Bammigatti prasannahb@gmail.com
Department of Computer Science and Engineering
S D M College of Engineering and Technology,
Dharwad, Karnataka, India

P R Rao pralhadrrao@gmail.com
Department of Computer Science and Technology
Goa University, Goa, India

Abstract

Role -based access control (RBAC) has been introduced in the last few years, and offers a powerful means of specifying access control decisions. The model of RBAC usually assumes that, if there is a role hierarchy then access rights are inherited upwards through the hierarchy.

In organization workflow the main threat is of access control. The Role based access control is one of the best suitable access control model one can think of. It is not only the role hierarchies but also other control factors that affect the access control in the workflow. The paper discusses the control factors and role hierarchies in workflow and brings a new model of RBAC. This paper also over comes the conflicts and proves that the system is safe by applying the new model to the workflow.

Keywords: RBAC, Control factors, Delegation.

1. Introduction

The concept of role is well known. Its standard definition [1] is "a job function within the organization that describes the authority and responsibility conferred on a user assigned to the role". The concept of role in access control is critical and efficient one [2]. The role was taken as the fundamental key component in the reference model proposed [1]. The factors that made role based access control to be used in workflow are [3]

Only a single rule can be applied, when there are multiple occupants of a single position

The access rules do not have to be changed when user's role is changed

Separation of duties policies can be enforced for conflicting roles which place constraints on concurrent role occupancy

In [1], the RBAC framework is extended to include role hierarchies. The model allows the occupants of superior roles to inherit all the positive access rights of their inferiors, and conversely ensures that the occupants of inferior positions inherit any prohibitions that apply to their superiors. However, the authors have pointed the situations that inheritance of access rights down the organizational hierarchy may be undesirable, and suggested the two possible ways of avoiding this by defining the entirely a new ordering of organizational hierarchy to define role hierarchy or defining subsidiary (private) roles outside the organizational hierarchy.

The new ordering is referred in the extended RBAC models like users' location context [4], time context [5], Task-role based access control (TRBAC) and coalition-based access control (CBAC)

[6]. The discussion in these is extended models limited to only two parameters and with fewer constraints.

The paper GenericWA-RBAC: Role Based Access Control Model for Web Applications [7] discusses the general scenario when users from different organization tries to access the native database and maps the different users according to the role hierarchy of the native system. The paper suffers seriously to assure the proper access rights to be assigned to the users of other organization requested for information.

The extended models referred above doesn't discuss about the fitness of the models for the workflow as there are other factors influencing in the workflow, as it is implementation of basic or extended RBAC models for workflow is dreadful.

Work In [8], Vieira and Rito-Silva propose the Work Analysis Refinement Modeling (WARM) methodology, a first approach to derive workflow process definitions by using business process models. This model was having serious drawback of concentrating only on the functional aspect of workflow and neglected the non-functional requirement mainly access control.

In the paper Workflow Access Control from a Business Perspective [9], the basic role based access model is taken for the access control in workflow. This model again doesn't discuss about the other affecting factors in workflow like delegation, decentralization and review.

The rest of paper is organized as follows: the section 2 discusses the control factors in the workflow section 3 narrates the relationship of role hierarchy with control factors; section 4 discusses the new RBAC model for the workflow considering the control factors. Section 5 includes conclusion and future work.

2. Control factors in workflow

Publicly quoted companies and government departments, and most large organizations publicize control factors, which apply throughout the organization. The usage of control factor is also becoming common practice in many systems development organizations and even quality standards recommends (ISO 9000 Standards series [4]) these factors to be mandatory. The control factors are requirements of regulators in the development of critical systems The UK defense, desires these factors for procurement of the safety critical system [10].

The general key components like users, roles, operations and objects are used for access control model and the relations are also quite well defined. The main relations defined in basic RBAC model are user assignment, permission assignments. In case of workflow the other factors also affect the access control. The control factors are discussed below.

1 Decentralization

In a very large organization it becomes impossible for a single person to carryout all the responsibilities assigned to him but in turn there is no option of skipping from his responsibilities. In such scenario the concept of decentralization comes into picture. The superior role will distribute the work responsibility to the some of junior roles. Then the junior roles will have full authority to carry out those actions. Such works are subjected to the review. The key points about the decentralization is

- By assigning authority to the junior role, senior role assign their own immediate access rights to carry out those actions.
- Senior role that have assigned the access rights, are not lost the ability to withdraw the assigned rights to junior role and either perform actions themselves or, to assign those actions to a different person of same role hierarchy or to different role all together [11].

2 Delegation

The decentralization goes hand in hand with decentralization. The senior role, distributes the responsibility to the junior roles and making the junior role equally responsible for the completion of the assigned responsibility by applying the supervision or through review. In the access control scenario the junior role accepts the rights the rights of senior role temporarily and performs the responsibility assigned by the senior role.

While assigning the permissions the senior role will allow the permission for the task to be performed by the junior role but care is taken to block the other activities, which can be performed by the newly assigned rights and they are the responsibilities of the senior role.

3 Supervision

There is of course a danger that delegates will not carry out their duties properly. For decentralization to work satisfactorily, an additional control principle is needed: supervision and review. This control principle requires one person's actions to be reviewed post hoc by another person, typically their superior in the position hierarchy. The superior usually does not exert direct control over the supervisee at the time that the actions are taken. Supervision is an activity that is carried out on someone by someone else in the immediately superior position. It consists of many activities including monitoring, appraisal, advising, praising and criticize, and outside the scope of any present-day access control system.

4 Review

Review, on the other hand, is carried out on specific activities. In the example that we give in section 3, there is a well-defined review activity for the Accounts Manager, which can be controlled by an access control system provided that it is carried out as part of a computerized application.

5 Separation of Duties

This control factor has been in existence and is familiar to the computer security community from the Clark-Wilson commercial security model [12]. Every critical transaction, the implementation is done by breaking the transaction into at least two separate actions. It is then required that the two actions should not be performed by the same person. This is very effectively implemented in role-based access control by defining mutually incompatible roles, with a constraint preventing their occupation by the same person, either simultaneously or in some time-related fashion [1]. Positive access rights for each of the actions are exclusively assigned to the two incompatible roles, and the constraint enforces separation of duties.

3. Relationship of Control factors in Role hierarchy

In the basic model of RBAC the entities involved in access control are users, roles, permissions, objects. The relationship UA (User assignment) maps the user directly to the hierarchy defined in organization. The hierarchy so far considered is similar to the organizational chart of organization or with little modification organization might have defined its role hierarchies. The important part RBAC relationship is PA (permission assignment) where in the specific permissions are assigned to roles to access the objects, a role can perform the specific task of querying the objects for specific required task assigned to role.

As in earlier section we have shown that in the workflow environment it is not only the strict hierarchy but also the control factors, which do effect the permissions dynamically for the specific role depending on the task. Before we take up the control factors in relationship of role hierarchies, we discuss the relations “is a”, which relaxes strict role hierarchy in the organizational workflow and the “part of” relation that uses the almost all control factors.

1 “is a” relation in role hierarchy

In the actual hierarchy there could be different roles of one kind of group eg. The organization may have different managers like project, technical manager, accounts manager etc., but there could be similar one task like billing each manager has to fill in. For the same reason, the role hierarchy includes the virtual role "manager" where in all the different managers belong to this role.

r1, r2 $\subset$ R (role)

r1 is a r2 $\rightarrow$ r2 specializes r1

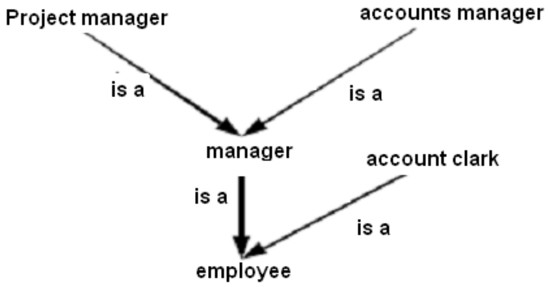

FIGURE 1: Example illustrating "is a" relation

In the above case "manager" is a virtual role. There is no user in the system belongs to only manager. All the managers in the organization are having the privileges of manager. Identification of such virtual roles deduces the redundancy in assignment of permissions to different roles and also the information is shared in secured manner.

2 "part of " relation in role hierarchy

"Part of" relation is known as aggregation relationship. The different roles are performing the task may involve control factors and these makes the relation complex, when designing the RABC considering the control factors.

A similar concept applies to the activities of an organization as illustrated in Figure 2, the Financial Control activity is composed of Financial Forecasting and Financial Accounting, etc, down to the Accounts Payable and Accounts Receivable activities. The activity hierarchy is partially ordered by subsets of activities.

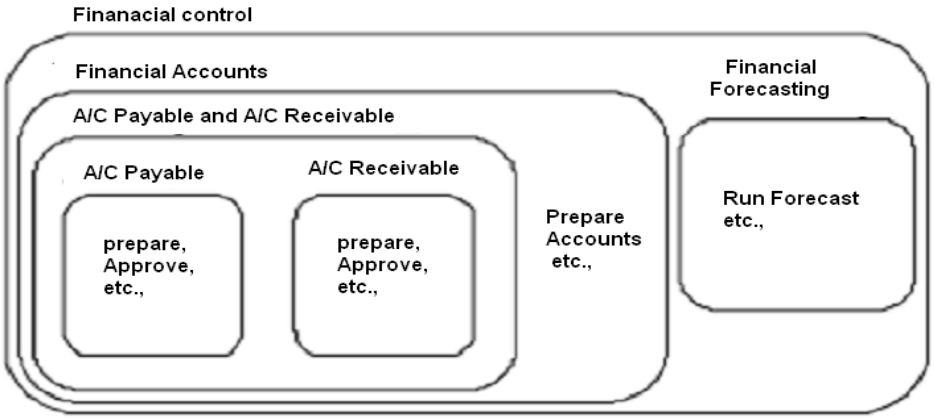

FIGURE 2: Hierarchy based on task uses "Part of" relations

The control factor delegation is effective, when there are more activities to be done by the role with proper supervision.

Let T = $\{T_1, T_2, T_3 \ldots T_n\}$ set of activities

$R = \{R_{1,}\ R_{2}, R_{3}\ldots R_{j}\}$ set of roles in organization

R_1 is responsiblefor T_1

R_1 does T_1 && R_2 delegates (R_1, T_1)

Then R_2 is responsiblefor T_1

In workflow of organization, the delegation is considered is achieved with strict role hierarchy for the above example in Figure 2. The one branch of role hierarchy is demonstrated in Figure 3.

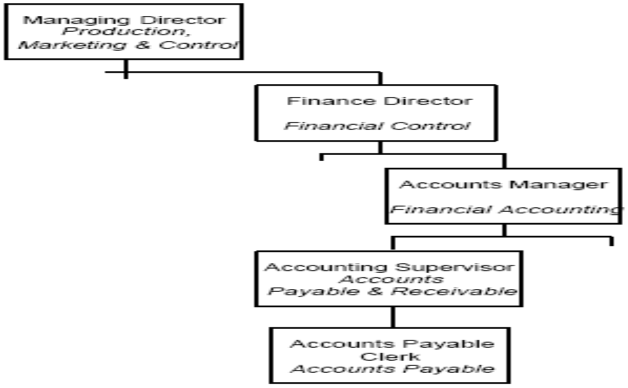

FIGURE 3: Role hierarchy for the delegation

3 Supervision in role hierarchy

The superior role delegates the activity to junior role. The senior is responsible for the task, which he has delegated. There requires the supervision in the process. The supervisor when delegating the task also assigns the required rights to carryout the task. Assignments of rights have to be done carefully otherwise there is a great access control flaw occurs. As far as the supervision is concerned it has to be strictly senior role in hierarchy.

The summary is represented as below.

Let Role R_i delegate the activity T_k to R_j then

R_i Senior to R_j and

Activity T_k is supervised by the role R_i or R_p (where R_p is senior or equal to role hierarchy R_i)

The Figure 4, illustrates the example for financial control activities of Figure 2 with activity role delegation and supervision with role hierarchy.

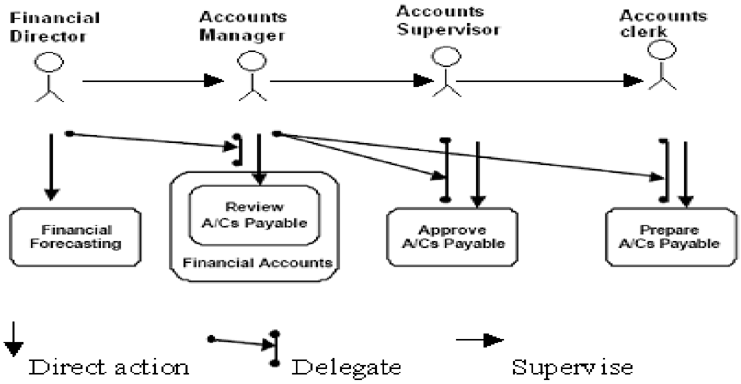

FIGURE 4: Illustration of delegation and supervision activities

The cheque issue activity with different roles involved is demonstrated as below and is self-explanatory.

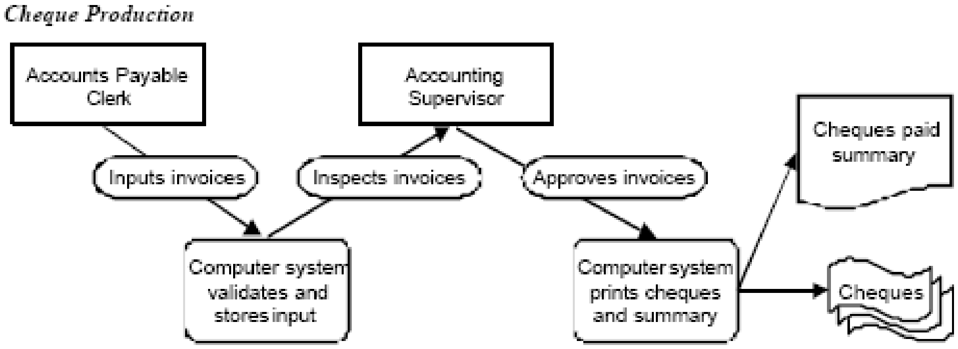

FIGURE 5: The cheque issue activity

4. Proposed Model

The new model proposed includes all the control factors and role hierarchy relationships discussed in section 2 and section 3. We use the role-based hierarchy defined originally in workflow and consolidate this hierarchy with aggregation and supervision hierarchy relationship.

1 Limitations of Existing model

The core RBAC model or any extension models are having the components user, roles, operations, objects and sessions with two relations defined as User Assignment (UA) and Permission Assignment (PA) [1].

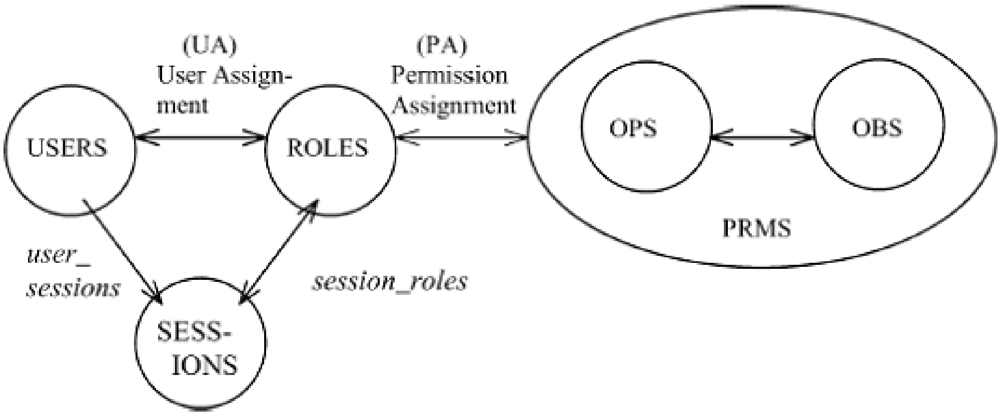

FIGURE 6: Core RBAC Model

The key points to be noted with Core Role Based Access Control model are
1 Set of users exists in organization
2 The roles are specified and these roles are in partial order of hierarchy

3 Each user is mapped to one or more role (many to many relationship) with user→ Role (UA) assignment relation

4 Each role is authorized with permissions to objects through operations on objects with Role → Permission (PA) assignment relationship

From the basic model it is evident that role performing the action/task (if authorized) is having the privileges on the objects, that were assigned to role. When considered the delegation control factor, superior role will delegate the task to one of it inferior role. i,e Superior role allots the specific grants to objects and assigns to a temporary role (but this role exists in hierarchy as junior role) to perform the task. If any of the control factors are introduced in the basic RBAC model, it fails, as there is no component for task/action with associated relationship exists in basic model. This leads model to fail miserably.

2 New Model

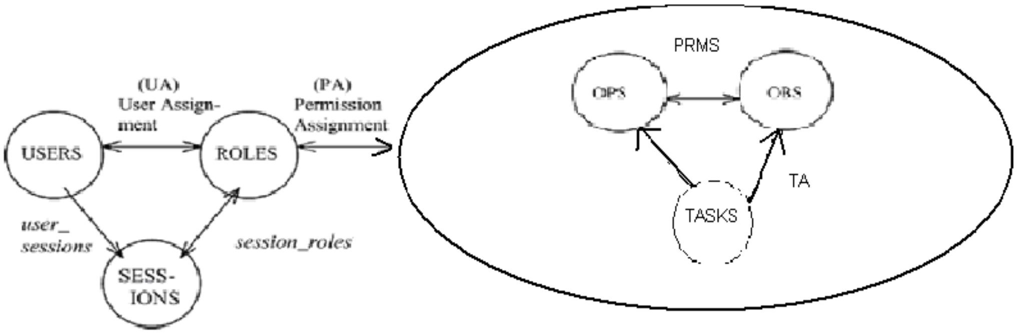

FIGURE 6: Proposed New Model

The new component TASKS is introduced in this model so that the delegation of task is possible from one role (superior role) to another (inferior role). When 'task' is accommodated as component in the model, one assignment relationship also get introduced as TA. The definition of basic components and relationships of new RBAC model are as defined below.

- USERS, ROLES, OPS, and OBS,TASKS (users, roles, operations, objects and TASKS respectively).
- UA $\subseteq$ USERS XROLES, a many-to-many mapping user-to-role assignment relation.
- assigned_users: (r:ROLES)→ 2^{USERS}, the mapping of role r onto a set of users. Formally: assigned users(r) ={ u$\in$ USERS | (u, r) $\in$ UA}.
- PRMS = $2^{(OPS\ XOBS)}$, the set of permissions.
- PA$\subseteq$PRMS X ROLES, a many-to-many mapping permission-to-role assignment relation.
- assigned_permissions(r: ROLES) → 2^{PRMS}, the mapping of role r onto a set of permissions. Formally: assigned permissions(r)= {p $\in$ PRMS | (p, r) $\in$ PA}
- Ob (p: PRMS) → { op $\subseteq$ OPS}, the permission-to-operation mapping, which gives the set of operations associated with permission p.
- Ob(p: PRMS) → { ob$\subseteq$ OBS}, the permission-to-object mapping, which gives the set of objects associated with permission p.
- SESSIONS, the set of sessions.
- user_sessions (u: USERS) → $2^{SESSIONS}$, the mapping of user u onto a set of sessions.
- session_ roles (s: SESSIONS) → 2^{ROLES}, the mapping of session s onto a set of roles.

Formally: session roles (si) $\subseteq$ {r$\in$ ROLES | (session users (si), r) $\in$ UA}.

- avail_session perms(s:SESSIONS) → 2^{PRMS}, the permissions available to a user in a session $\cup$ assigned permissions(r).
  r$\in$ session roles(s)

- TASKS=$2^{(OPS \ X \ OBS \ X \ ROLES)}$,the set of tasks
- task_assignment $\subseteq$ TASKS X ROLES, a many to many mapping of task-to-roles relationship
- assigned_tasks(r:ROLES) → $2^{(TASKS)}$,the mapping of role r onto set of tasks. Formally assigned _tasks→ {t$\in$TASKS | (t,r) $\in$ TA }

3 Basics of New Model

The new task-assignment relationship needs the powerful policy database, which verifies the authentication and authorization for the delegation of tasks from one role to another. The policy base maintains the information about Role, Objects, Tasks, Permissions and Attributes of objects. The following constraints have to be taken care by the policy base.

1 RULE 1

If only Objects are not available for role R_i (to whom the task is delegated) to perform the task t_i

then

Rj delegates ti → Ri (Role Rj delegates the Task ti to Ri)
Task ti involves extra objects {Ok, Ok+1, .. Ok+i}
Then PA relation assigns the objects to Role Ri
Ri -> {objects belong to Ri} + { Ok, Ok+1, .. Ok+i}
Ri is inferior and authorized to perform task therefore the system is safe

2 RULE 2

When Task is delegated already all objects are accessible by role Ri –Then no extra objects should be made available

(i) $\exists$
Ri objects→{Oi,Oi+1…Oi+k}
$\exists$Ri Task delegated t_i → {Oi,Oi+1…Oi+k}
No extra objects made available. System is safe but no guarantee system will run smooth that all objects are having permissions required by t_i

(ii) If $\exists$Ri Task delegated Ti objects available → {Oi,Oi+1…Oi+k}
$\forall$ ($\exists$Oi Verify p_i associated with Task Ti)

```
        if $p_i$ to be granted then
                store the $p_i$'s of $o_i$ for $R_i$
                grant new $p_i$'s = {original $p_i$ of $o_i$ } + { New $p_i$ for $o_i$ which tasks needs}
```

Now the system is live and safe

3 RULE 3

If not all objects for task Ti are available and not all permissions are associated with Objects

(i) For the first part Rule 1 can be applied (i.e. allotting the objects) here system is safe but no guarantee of liveness as permissions are not granted

(ii) For the second part of this scenario apply Rule 2 for all objects the system live and safe

4 RULE 4

If Objects are made available according to t_i Task requirement but not all the attributes of objects are made available

Create the new object with these attributes without violating the constraints on object assign to the role R_i once Supervision or review takes place the R_i (who delegated task) is append/updated the original Object

4 Design of New Model

In the new proposed model the essential components needed are policy base and a server. The server will authenticate and authorize the delegation of task and policy base will evaluate the constraints on the tasks with respect to objects, permissions and the roles.

The server requests for the data in the policy base, based on the request it got from role to delegation of work to other role. The main validations carried out by server is to verify whether the role requested for delegation of work is superior then to whom the work to be delegated. The server also validates the information about the objects, permissions to be associated to task after delegation.

The Policy base provides the information to the server about the objects and permissions needed by the tasks.

The Figure 7 illustrates the functionality of server and the policy base.

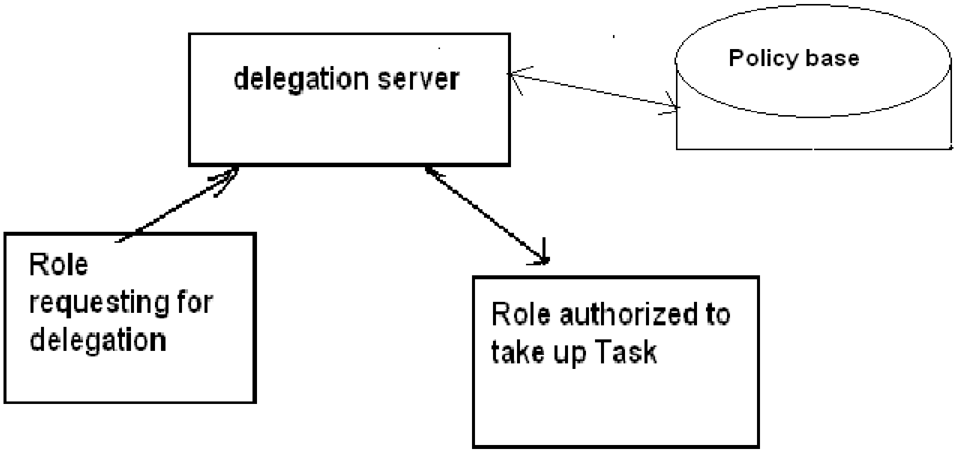
delegation server
Policy base
Role requesting for delegation
Role authorized to take up Task

FIGURE 7: Delegation Server and Policy base

5. Conclusion and Future work

The delegation control factor plays a very key role in the workflow system. The implementation of delegation can be adopted in Role Based Access Control model, in any other access control model the implementation becomes very difficult. The proposed new model for the RBAC allows the inclusion of delegation control factor. The efficient design of policies makes it stronger and provides the easy of work with safety. The policy base and server accommodates the task assignment relationship.

The idea of the delegation in RBAC can also be enhanced to the other two control factors, supervision and the review. The same policy base and servers can be strengthened to incorporate these two control factors. The new rules also can be formed to make the workflow system to operate in safe and live conditions.

Acknowledgement

Authors proudly acknowledge Dr.Srinath Srinivasa for his kind support and cooperation extended during this work.

6. References

1. Sandhu, R.S., et al., "*Role-Based Access Control Models.*" IEEE Computer, 1996. 29(2): p. 38-48.
2. Ting, T.C., *"A User-Role Based Data Security Approach, in Database Security: Status and Prospects,"* C.E. Landwehr, Editor. 1988, Elsevier.
3. Jonathan D. Moffett., *"Control Prnciples and Role Hierarchies"* in Proc of 3rd ACM Workshop on RBAC oct 1998 No. 3, August 2001, pp. 224-274.
4. Bertino, E., Damiani et al., *"GEO-RBAC: A Spatially Aware RBAC"*. in 10th Symposium on Access Control Models and Technologies (SACMAT'05), (2005).
5. Joshi, J.B.D., Bertino, E., et al.,. *"A generalized temporal role-based access control model"* IEEE Transactions on Knowledge and Data Engineering, 17 (1). 4-23, 2005
6. E. Cohen, R. K. Thomas, W. Winsborough and D. Shands. *"Models for Coalition-based Access Control (CBAC)"* Seventh ACM Symposium on Access Control Models and Technologies. 2002, Monterey, California, USA
7. Prasanna H B, P R Rao *"GenericWA-RBAC: Role Based Access Control Model for Web Applications"* IEEE Conference ICIT Dec 2006
8. P. Vieira, and A. Rito-Silva, *"Work Analysis Refinement Modeling"*, INESC-ID Technical Report, 2003
9. Dulce Domingos et al., *"Workflow Access Control from a Business Perspective"* In Proceedings of the 6th International Conference on Enterprise Information Systems (ICEIS' 04)
10. DEFSTAN 00-55, *"The Procurement of Safety Critical Software in Defence Equipment: Part 1 Requirements & Part 2: Guidance."* UK Ministry of Defence, 1 August 1997.
11. Moffett, J.D. and M.S. Sloman, *"Delegation of Authority, in Integrated Network Management II, I"* Krishnan and W. Zimmer, Editors. 1991, North Holland. p. 595-606
12. Clark, D.C. and D.R. Wilson. *"A Comparison of Commercial and Military Computer Security Policies."* in IEEE Symposium on Security and Privacy. 1987. Oakland, CA: IEEE Computer Society Press.