Proceedings of the 3rd ACM Workshop on Role Based Access Control (RBAC), 22-23 October 1998
George Mason University, Fairfax, VA. ACM Press. ISBN 1 58113 113 5

Control Principles and Role Hierarchies

Jonathan D. Moffett
Department of Computer Science, University of York
Heslington, York YO1 5DD, UK
jdm@cs.york.ac.uk

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. This paper examines the relationship between the inheritance properties of role hierarchies and control principles which are used in many large organisations: separation of duties; delegation; and supervision and review. It discusses possible relationships between roles and identifies three different kinds of role hierarchy. The control principles and role hierarchies are illustrated in a realistic application, and their interactions are discussed. It emerges that there may be conflict between control principles and the inheritance of access rights through a role hierarchy. Some ways in which role hierarchies can be used for safe inheritance of access rights are discussed.

1. Introduction

The concept of role is well-established in the literature of sociology. Its standard definition [1] is the set of rights and duties associated with a position, which are assigned to a person who occupies that position. Consistent with this is the definition given in [2]: "a job function within the organization that describes the authority and responsibility conferred on a user assigned to the role".

The value of roles for access control has been known for some time, e.g. [3]. Their use was given a new impetus by the paper on Role-Based Access Control (RBAC) Models [2] which proposed a framework of reference models for role-based access control. The motivating impetus was the intuition that, in most organisations, access control decisions are based upon the appropriate role for the performance of actions, and not upon individual people. This can be seen to have great advantages:

?? Only a single rule needs to be made when there are multiple occupants of a single position;

?? The access rule do not have to be changed when there is a change in the occupant of a position;

?? Many policies for separation of duties can be enforced by declaring conflicting roles which place constraints on concurrent role occupancy.

In [2], 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 of that paper observe that in some situations inheritance of access rights down the organisational hierarchy may be undesirable, and outline two possible ways of avoiding this:

?? Using some other ordering than the organisational hierarchy to define the role hierarchy; or

?? Defining subsidiary ("private") roles outside the hierarchy.

In this paper we aim to take this discussion further by outlining the control principles which are applied in many large organisations and their impact on inherited access rights, reaching some conclusions about the appropriate use of inheritance and hierarchies.

We approach this as follows: section 2 introduces some common Organisational Control Principles; section 3 discusses Roles and their Relationships. Section 4 uses an example to motivate the discussion, and section 5 discusses the interaction of control principles and role hierarchies, and the consequences for access control. Finally, section 6 reaches some conclusions.

2. Organisational Control Principles

Most large organisations, such as publicly quoted companies and government departments, promulgate control principles which apply throughout the organisation. This practice is also becoming common practice in systems development organisations, under the influence of the drive for quality standards (ISO 9000 Standards series [4]) and the requirements of regulators in the development of critical systems, e.g. the standards laid down for procurement of safety critical software in defence equipment in the UK [5].

The following control principles are in use in at least two very large commercial firms that are known to the author:

Separation of duties. This control principle has been in existence for upwards of a century in financial organisations and is familiar to the computer security community from the Clark-Wilson commercial security model [6]. It is normally defined for critical transactions and is implemented 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 elegantly 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 [7]. Positive access rights for each of the actions are exclusively assigned to the two incompatible roles, and the constraint enforces separation of duties.

Decentralisation. This control principle recognises that, in a large organisation, it is impossible for one person to manage directly all the activities of the organisation. Therefore, some activities are **delegated** to people in inferior positions – we will refer to them as "delegates" (noun). They then have full authority to carry out those actions, though they are normally subject to supervision and review from their superiors. Note two points about this principle:

?? By delegating authority to the delegate, delegators abrogate their own immediate power to carry out those actions, otherwise the purpose of decentralisation would be partially frustrated;

?? In spite of their abrogation of their direct ability to take the actions which have been delegated, delegators have not lost the ability to withdraw the delegation and either perform actions themselves or, more likely, delegate those actions to a different person. There are no difficulties raised by delegators removing access rights from themselves, as they can subsequently restore them if necessary [8].

Again this is elegantly implemented by roles, by the following steps:

a) Giving permissions for the activities in the delegate role;

b) Removing the permissions from the delegator.

Delegation can be transferred by allocating a different person to the delegate position, or withdrawn by reversing the steps.

Supervision and review. There is of course a danger that delegates will not carry out their duties properly. For decentralisation 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 admonishing, and outside the scope of any present-day access control system.

Review, on the other hand, is carried out on specific activities. In the example that we give in section 4, 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 computerised application.

3. Roles and their Relationships

We informally regard roles, positions, actions, etc, as objects. We examine their possible relationships using the framework for object modelling defined in [9]. A link is a connection between object instances, and an association describes a group of links with common structure and semantics. The link may be traversed in either the forward or the inverse direction. When using formal specification, associations may be defined as relations, with the inverse direction being defined by the inverse relation. Two specific kinds of association are sufficiently important to have generic names: Generalisation and Aggregation.

There has been an implicit assumption in the earlier literature that there is only one useful kind of role hierarchy. We identify here three candidate role hierarchies:

?? The *isa* role hierarchy, based on generalisation;

?? The Activity role hierarchy, based on aggregation;

?? The Supervision role hierarchy, based on the organisational hierarchy.

We do not claim that this classification is exhaustive; there may be other useful role hierarchies.

3.1 Generalisation: the "isa" hierarchy

[2] gives an example of generalisation, also known as the "isa" relationship:

A PrimaryCarePhysician *isa*
Physician *isa* HealthCareProvider

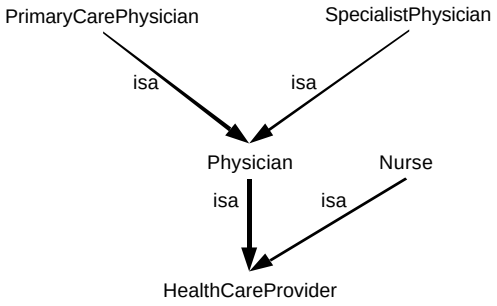

Figure 1 A Role Hierarchy Based on Generalisation

Each of these roles is more general than the previous one, and they constitute a partial order. Traversing in the inverse direction we have specialisation: PrimaryCarePhysician specialises Physician specialises HealthCareProvider. See also figure 1, extended from [2]. Formally:

R is the set of all roles.

isa, specialises: R ? R

Both *isa* and *specialises* are strict partial orders: irreflexive, antisymmetric and transitive (formal predicates omitted). The two relations are the inverse of each other.

? r1, r2 ? R ? r1 isa r2 ? r2 specialises r1

We further make the assumption, for use in later discussion, that some of the roles in the *isa* hierarchy may be virtual, i.e. no user occupies them; they are only defined to capture qualities which are shared by real roles further up the *isa* hierarchy. No virtual role may be above a real role in the hierarchy.

U is the set of users.

occupies: U ? R

virtual: R

? r ? R ? virtual (r) ? ? ? u ? U ? u occupies r

? ? r1, r2 ? R ?
r1 ? virtual ? ? (r2 ? virtual) ? r1 isa r2

In figure 1 Physician and HealthCareProvider are virtual roles: Physician captures the commonality between PrimaryCarePhysician and SpecialistPhysician; while HealthCareProvider captures the commonality between Physician and Nurse.

3.2 Aggregation: the Activity Hierarchy

Aggregation is also known as the "part of" relationship; complex objects are composed of, or aggregated from, parts. A similar concept applies to the activities of an organisation as illustrated in figure 2: the Financial Control activity is composed of Financial Forecasting and Financial Accounting, etc, etc, down to the Accounts Payable and

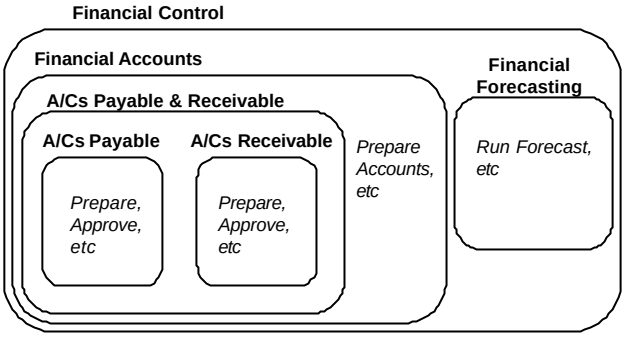

Figure 2 A Hierarchy Based on Aggregation

Accounts Receivable activities. The activity hierarchy is partially ordered by subsets of activities.

It is possible to define a role hierarchy based on activities. Given a set of activities A we can define *ResponsibleFor* and *Does*, which are relationships between roles and sets of activities. If a role is responsible for an activity, either it does it directly, or it *Delegates* responsibility for it to another role:

A is the set of direct activities carried out, e.g.
PrepareA/CsPayable

ResponsibleFor, Does: R ? ? A

Delegates: R ? R ? ? A

? r1 ? R, A1 ? ? A ? r1 ResponsibleFor A1 ?
(r1 Does A1 ? ? r2 ? r1 Delegates (r2, A1) ?
r2 ResponsibleFor A1)

We can induce a partial order on roles, for which we use the descriptive but awkward phrase *MoreActivitiesThan*, via the activity hierarchy. We order roles on the activities for which they are responsible, as follows:

MoreActivitiesThan: R ? R

? r1, r2 ? R, A1, A2 ? ? A ?
r1 ResponsibleFor A1 ? r2 ResponsibleFor A2 ?
r1 MoreActivitiesThan r2 ? A1 ? A2

It is a strict partial order because it is based on the subset relation between activities. We will describe this ordering as the Activity Hierarchy; the higher up the hierarchy, the greater the number of activities for which a role is responsible. So, from figure 2, we can see that FinancialControl has *MoreActivitiesThan* AccountsPayable. In the example that we are using, the activity hierarchy is identical with the organisational hierarchy (see below and figure 3), but there is no general reason why this should be so; responsibility may be delegated out to a different part of the organisation or contracted out.

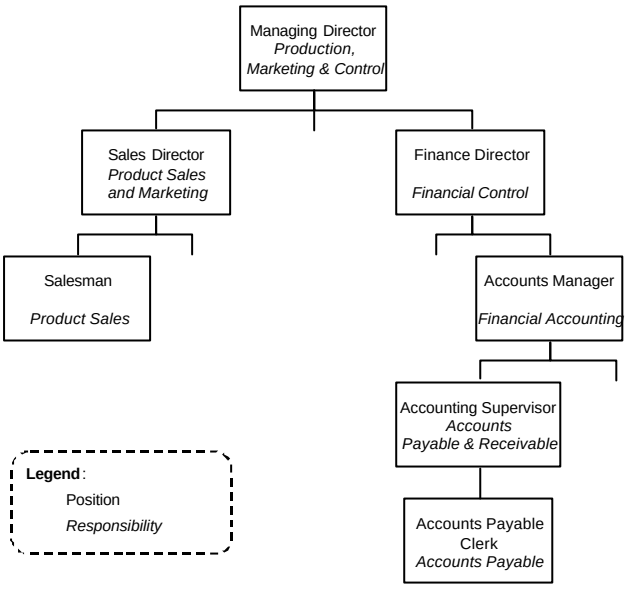

Figure 3 An Organisation Chart

3.3 Supervision Hierarchy

Most formal organisations describe their fixed positions by means of an organisation chart, which describes a strict partially ordered set of named positions. An example is shown in figure 3. It is in the form of a rooted tree, with the root at the top of the organisation. Each position has one or more roles:

?? The set of activities for which the position is responsible, shown in the figure in italics, e.g. the Finance Director has the role of Financial Control;

?? The Supervisor role for immediate inferiors in the hierarchy – it is this relationship which usually defines the hierarchy in the first place;

?? The Review role for activities which this position is required to review. We mention this role because of its relevance to control principles, but it does not form the basis of a hierarchy; it is often, but not necessarily, carried out by the immediate superior in the hierarchy.

Without attempting to define the semantics of supervision formally, we can define the supervision hierarchy via the Supervises and IndirectSupervises relations:

P is the set of positions.

Supervises, IndirectSupervises: P ? P

? p1, p2 ? P ? p1 IndirectSupervises p2 ?
p1 Supervises p2 ? ? p3 ? P ?
(p1 Supervises p3 ? p3 IndirectSupervises p2)

IndirectSupervises is a strict partial order because of its derivation from the organisation chart. It is, of course, a position hierarchy, not a role hierarchy, but the RBAC

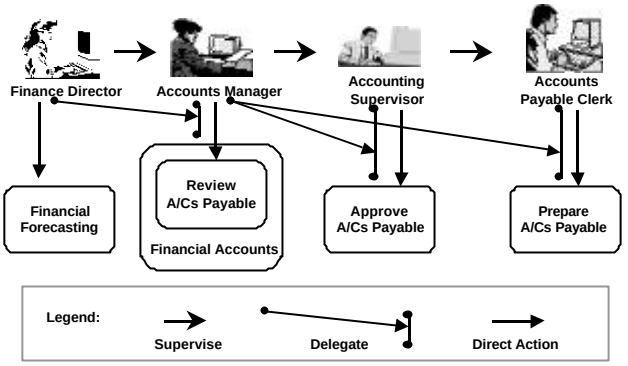

Figure 4 Financial Control Activities

literature has not previously distinguished between roles and positions, so we include it here.

4. A Motivating Example

In order to illustrate our points we discuss the Organisation Chart shown in figure 3 and some of the activities that are associated with it. The figure shows the hierarchy of positions with supervisory responsibility denoted by a downward connecting line, and each position has an associated area of responsibility, shown in italics.

As an example, the Finance Director is responsible for all of Financial Control. Financial Forecasting is a direct activity of the role. The following activities are delegated (a fragment of the whole).

?? Financial Accounts to Accounts Manager

?? *Review* Accounts Payable *to* Accounts Manager

?? *Approve* Accounts Payable *to* Accounting Supervisor

?? *Prepare* Accounts Payable *to* Accounts Payable Clerk

Figure 4 shows the delegation, supervision, review and other activities in this area:

4.1 The Accounts Payable System

We take this example further to illustrate the distinction between approval and review. Figure 5 shows the Accounts Payable System. There are two stages to it:

Cheque Production

In order to produce cheques to pay the organisation's invoices:

?? The Accounts Payable Clerk *Prepares* Accounts Payable by inputting details of invoices to the system, which validates and stores the details;

?? The Accounting Supervisor *Approves* Accounts Payable by inspecting the details which have been input and if approved, releasing them for payment, as a result of which the system prints the cheques themselves (or perhaps perform electronic funds

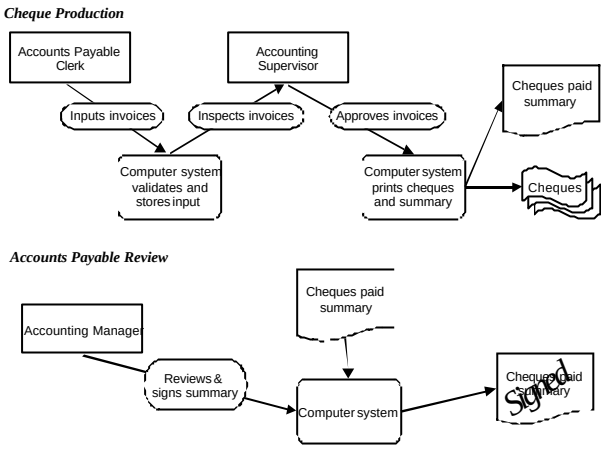

Figure 5 Accounts Payable System

transfer) and outputs a summary of what has been done.

The Approval action is an integral part of the transaction; until it has been carried out the system cannot release funds, and after it has been carried out successfully nothing can stop the flow of funds.

Accounts Payable Review

Although the requirements of separation of duties have been achieved in the first stage, the Accounts Manager is still required to *Review* the transactions at some later time to be satisfied that they have been carried out successfully, and the Auditors will inspect for the Manager's signature on the summary as evidence that this review has been carried out. In many organisations today, the summary is printed out on paper and the signature is in ink on paper, but the technology is available for digital signatures on an electronic copy and then the access control system can control the digital signature. Review is, of course, a *post hoc* activity which cannot stop the funds but could discover that some irregularity has occurred.

Access Rights in the Accounts Payable System

There are three sets of actions requiring permission:

?? Prepare invoices. The Accounts Payable Clerk needs permission for this;

?? Approve invoices. The Accounting Supervisor needs permission for this;

?? Review invoices paid. Accounts Manager needs permission for this.

Apart from these permissions, all subjects are prohibited from carrying out any of the actions.

The roles of Preparation, Approval and Review are made mutually incompatible, so that no position can be associated with more than one of them.

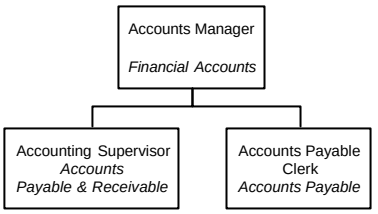

Figure 6 Fragment of Alternative Organisation Chart

4.2 Role Hierarchies in the Organisation

We defined three kinds of role hierarchy above:

?? The Supervision hierarchy is clearly defined by the organisation chart, figure 3.

?? The Activity role hierarchy is also defined by the organisation chart, as each position has a responsibility role associated with it, and the responsibilities are ordered down the chart by the subset relationship, as is the case in many organisations.

?? The *isa* role hierarchy is not defined for our example.

5. Discussion

5.1 Access Right Inheritance and the Control Principles

In this section we examine what effect the inheritance of access rights through the organisation hierarchy would have on the maintenance of control principles in the organisation.

Separation of duties: In the absence of access right inheritance, separation of duties can been achieved in the Accounts Payable system, because preparation and approval of the payments are carried out by two different, mutually exclusive, roles. However, if a simple Access Right Inheritance paradigm is applied, using either the Supervision hierarchy or the Activity role hierarchy, this separation is destroyed because the Supervisor would inherit the Clerk's rights, as the immediate superior in the hierarchy. The situation would be no better if this hierarchical relationship were broken by an alternative organisation structure, as in figure 6.

Although in figure 6 the Supervisor has not inherited the access rights of the Clerk, the Manager has inherited the access rights of them both.

Similar problems are introduced if there is an *isa* hierarchy defined, through which access rights are inherited, e.g. if by reason of professional competence the Accounting Supervisor *isa* Accounts Payable Clerk.

If inheritance of access rights is in operation, there are two possible approaches to ensuring that separation of duties is not violated:

?? Define the role hierarchy to avoid the problem. The practicality of this will vary from case to case; or

?? The access control system could prevent this violation of separation of duties if it prohibits the inheritance of access rights from conflicting roles, at the expense of increased complexity.

Supervision and review: A similar problem applies to the Manager's duty to Review the Accounts Payable results at a later time. That review cannot be regarded as impartial if it is possible that the Manager, having inherited access rights from lower in the hierarchy, may have participated in the activity which is to be reviewed.

***Decentralisation*:** The purpose of decentralisation, too, is undermined if, having delegated their powers, managers can still exercise them themselves through inheritance.

The general conclusion that we reach is that the control principles may be violated as a result of access right inheritance if an inappropriate role hierarchy is defined.

Further complexity in the access control system might enable maintenance of the control principles in spite of inheritance of rights. We have not examined this in detail, and in practice, it is likely to be infeasible; we have actually simplified the requirements of the control principles of a large organisation, and whatever additional controls were incorporated into the access control system would almost certainly turn out to be inadequate. We prefer an approach such as in [7], in which a logical language allows the construction of very flexible control policies.

5.2 The Uses of Access Right Inheritance

In this section we discuss the circumstances in which access right inheritance has a legitimate use.

Virtual Roles

There is a further example of role hierarchy given in [2], and reproduced in figure 7. In this figure, we believe that the hierarchy is an *isa* hierarchy, and that Project Member is a virtual role, constructed to capture the commonality between Test Engineer and Programmer. Its only purpose is to contain qualities that are to be inherited, and it appears to us always to be safe to inherit up an *isa* hierarchy from a virtual role, assuming that: compatibility constraints are only placed between real roles; and only real roles are supervised or act as delegates. There is, of course, as pointed out in that paper, danger in inheritance by the Project Supervisor from the Test Engineer or Programmer, which are real roles, and an alternative mechanism – private roles outside the role hierarchy – is suggested.

It would appear to be straightforward to enhance an access control system to distinguish between real and virtual roles, and always permit inheritance from virtual roles.

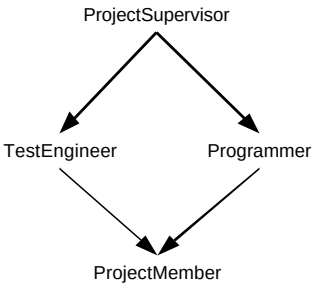

Figure 7 A Hierarchy of Real and Virtual Roles

Read Access

The examples that we have given have been concerned with maintaining the *integrity* of commercial transactions. When it comes to *confidentiality*, the integrity control principles do not apply. As a general principle, it could be organisational policy that anyone in the hierarchy should be able to read any document which can be read by their inferiors.

The main problem with this principle appears to us to be that the organisation needs to guard against a superior reading a draft document that is in preparation, and copying it before it has been completed to the inferior's satisfaction, which would effectively cause the integrity of the document to be violated. Even this can be dealt with in a principled fashion. There could be a policy that Read access rights inherit upwards, provided that the inferior's right is Read-only.

There would of course need to be *ad hoc* exceptions to this policy, both to guard an inferior's privacy rights in well-defined circumstances (e.g. personal appraisal documents in organisations where the appraisal is done out-of-line) and to ensure that the inferior does not hide illicit material.

Organisational Styles

The organisational style which leads to the imposition of control principles such as we have outlined is near-universal in well-established bureaucratic organisations which deal in valuable assets such as money. However, there may be other organisations for which access right inheritance is appropriate, e.g. expert-led organisations where the boss is very proactive and is not constrained by control principles.

Ad Hoc Exceptions

All writers advocating inheritance of access rights have also emphasised the importance of being able to override inheritance. The suggested method in [2] is the use of private roles which are outside the inheritance hierarchy. Other suggested methods [10] include the use of explicit

prohibitions which have priority over permissions, and the ability to limit the depth/height of inheritance.

All *ad hoc* exceptions suffer from the disadvantage that they obscure the clarity of a simple access control system and make access control administration and auditing more difficult. The question always has to be asked: is it easier to administer an unsophisticated system with few exceptions, or a powerful system whose elegance is spoilt by exceptions? Each organisation is likely to reach a different trade-off.

Inheritance of Delegation

One form of inheritance which could be considered is the downwards inheritance of delegation through the Activity Hierarchy. We observe that all access rights are propagated down from the top of the organisation by delegation. If the concept of delegation were "hard-wired" into the access control system, then the following *downwards* inheritance principle could be used, with no danger to the control principles:

> For any roles R1, R2 and action A, if R1 has *MoreActivitiesThan* R2 and R1 has the permission Delegate(A), then R1 is permitted to give R2 exactly one of the permissions Delegate(A) or A.

This enables the action A to be delegated as far down the hierarchy as wished, without any danger that unexpected violations of separation of duties will occur, although the normal constraints on incompatible role occupancy would still be required.

6. Conclusion

This paper has examined the concept of inheritance of access rights through a role hierarchy, and concluded that role hierarchies are less simple than they seem on first examination. As a result, inheritance schemes, unless they are carefully thought out, may be in conflict with the control principles which are in operation in many large organisations. We have also looked at some ways in which the problems could be mitigated.

The biggest difficulty appears to us to be that an organisation may have defined a role hierarchy for some other purpose and introduces it into the access control system without considering its impact on access rights. Anyone intending to use an access control system which incorporates inheritance of access rights through a role hierarchy should look closely at its impact on their system of management control.

Acknowledgements

We acknowledge the help of Dirk Jonscher in several useful discussions on this subject, and the anonymous reviewers, whose helpful comments have led to several improvements.

References

1. Thomas, E. and B. Biddle, *The Nature and History of Role Theory*, in *Role Theory: Concepts and Research*, B. Biddle and E. Thomas, Editors. 1979, Krieger Publishing.
2. Sandhu, R.S., *et al.*, *Role-Based Access Control Models.* IEEE Computer, 1996. **29**(2): p. 38-48.
3. Ting, T.C., *A User-Role Based Data Security Approach*, in *Database Security: Status and Prospects*, C.E. Landwehr, Editor. 1988, Elsevier.
4. ISO 9000, *Guidelines for the Selection and Use of Standards on Quality Management, Quality System Elements and Quality Assurance.* International Standards Organisation.
5. 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.
6. 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.
7. Jajodia, S., P. Samarati, and V.S. Subrahmanian. *A Logical Language for Expressing Authorizations*. in *IEEE Symposium on Security and Privacy*. 1997. Oakland, CA: IEEE Computer Society Press.
8. 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.
9. Rumbaugh, J. and M. Blaha, *Object-oriented modelling and design*. 1991: Prentice-Hall.
10. Jonscher, D. and K.R. Dittrich. *Argos - A Configurable Access Control System for Interoperable Environments*. in *Database Security, IX: Status and Prospects*. 1996: Chapman & Hall.