

FRAMEWORK FOR AGENT-BASED ROLE DELEGATION

EZEDIN BARKA

College of Information Technology
University of the United Arab Emirates
P.O.BOX 17555
Al Ain, Abu-Dhabi
UNITED ARAB EMIRATES
ebarka@uaeu.ac.ae

RAVI SANDHU

Information and Software
Engineering Department, MS 4A4
George Mason University,
Fairfax, VA 22030,
USA
Sandhu@gmu.edu

Abstract— This paper describes a framework for addressing the administration of role delegation introduced in the well-known role-based access control model (RBAC). More specifically, this paper describes how a third party, called an agent, can administer the delegation of roles on behalf of a user who is a member of a certain role and wishes to delegate his role to another user who belongs to another role. Furthermore, this paper describes a framework of reference to systematically address the diverse manifestations of the agent-based delegation, such as Role participant agent, Non-role participant agent, Static, and Dynamic types of delegation and introduces an agent-based delegation model that illustrates delegation based on non-role participant delegation.

Index Terms— Role-Based Access Control, Role-Based Delegation Model, Role Delegation, Role Revocation

I. INTRODUCTION

The framework for role-based delegation models (RBDM) described by Barka and Sandhu [7] has identified some interesting cases that showed that human-to-human delegation can occur in many different ways: Permanent, Temporary, One step, Self-acted, etc. Barka and Sandhu have also developed some models for delegation, in flat roles [7] and in hierarchical roles [BS2004], where the delegating user himself always administers the delegation.

This paper focuses on agent-based delegation, which addresses how a third party, called an agent, can administer the delegation on behalf of a user who is a member of a certain role and wishes to delegate his role to another user that belongs to another role. There are many manifestations in which agent-based delegation can be done in the role-based delegation: Role participant agent

Non-role participant agent, Static, and Dynamic types of delegation. In this paper, we describe a framework of reference to systematically address the diverse manifestations of the agent-based delegation model. We describe an agent-based delegation model that illustrates delegation based on non-role participant delegation. Other models to illustrate other manifestations can be similarly developed. Thus, we will be discussing them in less detail.

Agent-based delegation is motivated by the fact that in real

life, there are cases where a user who is a member of a certain role in an organization needs to delegate his/her role to some other user who is a member of a different role to complete some task. This can be accomplished using a third party (an agent) whose responsibility is only to administer the delegation between users in different roles, in the cases when the actual delegating role member is not available. For example, in physical space, a secretary of a department in a hospital can be given the keys to all physicians' offices, so that if any of the physicians is not available to complete a certain task, and someone else (such as a resident or a nurse) needs access to one of the physician's rooms in order to complete the task on behalf of the physicians, the secretary can administer the delegation for the physicians and give the key to the delegate to access the physician's room. In cyberspace, where the access to the system resources is controlled by the role memberships, an agent role can be defined to administer the delegation of that same physician's role to someone else in order to complete a task on behalf of the physician.

In this Paper, delegation will be addressed based on both the flat and the hierarchical relations between roles. First, we introduce the agent-based delegation with respect to the flat roles, and then, we will proceed with the agent-based role delegation where the relation between the roles involved is hierarchical.

There are self-delegations addressed in the literature, where a user can go out and grab a membership in a role to accomplish a certain task and then relinquish that membership once the task is completed. We consider this type of delegation as dangerous, simply because it lacks the element of control, which ensures accountability and traceability. We consider this type of delegation to be more serious if it is allowed to be propagated. Our approach treats delegation a little tighter than that of the self-delegation, by adding some restrictions to the classical discretionary delegation.

This type of delegation has not received much attention in the literature so far, and thus, it can be considered as the first attempt toward addressing the problem of an agent-based delegation between humans using roles. We emphasize that the delegation itself occurs within the computer system even

though it is human to human.

The rest of the paper is organized as following:

Section 2 introduces the agent-based role delegation with respect to flat roles, sub-section 2.1 describes a framework for the agent-based role delegation model, and sub-section 2.2 describes an agent-based delegation model based on the dynamic non-role participant agent (ABRDM-DNRP). Section 3 discusses the agent-based delegation in the cases of hierarchical roles. Finally, section 4 concludes this paper.

I. AGENT-BASED ROLE DELEGATION IN FLAT ROLES

This section addresses the agent-based delegation with respect to a flat relation between the roles involved in delegation. Flat roles means that the relationship between the roles inside the organization is lateral, and there is no hierarchical structure. This also means that there is no inheritance of permissions between the roles involved.

We start this section by developing a framework for handling the different manifestations of agent-based role delegation and then we develop a model to show how to implement different policies related to this type of delegation.

A. Framework for Agent-Based Role Delegation in flat roles

Our framework identifies two different manifestations in which an agent-based role delegation can be done: Role-participant agent and Non-role participant agent. These two manifestations can be further extended to include dynamic and static types of delegations. Table 1 depicts these manifestations.

Table 1

Framework for Agent-Based Role Delegation

	Role Participant Agent	Non-Role Participant Agent
Dynamic Delegation	ABRD-DRPA	ABRD -DNRPA
Static Delegation	ABRD -SRPA	ABRD -SNRPA

A key element in the agent-based delegation model is that the agent cannot have more power to delegate than the delegating user. Moreover, with agent-based delegation, we want to be able to control what can be delegated and to whom.

The following subsections describe the above mentioned manifestations:

1. Delegation by Role-Participant Agent

Role participant agent refers to the case where an agent who is a third party is assigned to administer the delegation between two different users that belong to two different roles, and that

agent has membership in the delegating role. This means that the middleman “agent” has full power in the delegating role (see figure 1). We consider this type of delegation as a restricted two-step delegation. A user who wishes to have a third party administer his role delegation can accomplish his wish by delegating his role to an agent with the authority to further delegate that role to another user that meets a criteria, qualifying him to a delegate user. For example, if a physician in a hospital elects to delegate his role to another employee in that hospital, that employee must be of a special status, i.e. another physician or a resident, and must be from the same department as that of the delegating physician. Furthermore, the delegation administered by the assigned agent cannot be further delegated. This restriction insures accountability and traceability.

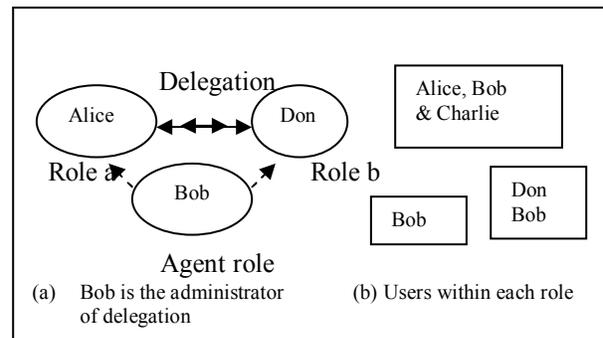


Fig. 1: Example of Non- Role Participant Agent Delegation

• Occurrences of Role-Participant Agent Delegation

The role-participant agent type of delegation can occur in two ways: statically or dynamically. In the static role-participant agent delegation, the delegating role member delegates his role membership to a user who is a member of a predefined role (agent role) for the purpose of further delegating that role to another specified user. For example, in Figure 1, Bob is designated by the security officer as the agent who administers the delegation on behalf of any other member in role a. In physical space, a physician in a hospital may give the key to his office to the department secretary, who is also a permanent member of an agent role, before he/she becomes unavailable to carry out a certain task, with the authority to further give that key to other residents or associates. In this case, the secretary obtains the power to enter and act in the physician’s office, with the assumption that the secretary is trusted not to misbehave in the physician’s office.

In the case of dynamic delegation, the delegating role member can, dynamically, delegate his role to another user who meets a certain criteria “set by the security officer,” with the authority to further delegate that role. For example, a physician who is not available to carry out a certain task and wishes to delegate that responsibility to his resident can call on someone else that he trusts to provide his resident access to his office in order to complete that task on his behalf.

The main difference between the two above-mentioned approaches is that, in the first case, a security officer statically assigns the user to be a member of the pre-defined agent role,

with the authority to act in all other roles. Therefore, that agent has full power in the delegating role and is trusted to delegate to any other user. So, when a user who belongs to one role needs to access another role in order to complete some task, that user can ask the agent directly for access to the second role.

In the second case, dynamic delegation, the delegating role member assigns the member to the agent role on the fly. That agent role member becomes a delegate member of the delegating role, hence obtaining the power of that role, and is trusted to further delegate that role to a third user based on the original user's request.

The following are some assumptions and basic elements that are specific to the role participant-agent mode of delegation:

- The delegation is restricted to two step delegation. This means that the role, which is delegated by the agent, cannot be further delegated.
- In both cases of the agent-based delegation, the agent can delegate on behalf of the delegating role member only.
- The agent can delegate only to roles that are defined by the delegating member.
- The agent can delegate the role memberships of some users to others but cannot delegate to himself, because it is meaningless.
- The delegation addressed in this model is temporary, which means that the delegating user can revoke the delegation at any time.

2. Delegation by Non-Role Participant Agent

The other manifestation of agent-based role delegation is non-role participant agent delegation. In this type of delegation, the agent is responsible only for administering the delegation on behalf of the delegating role member. Also, the agent does not possess any power to operate in the delegated role (Figure 2).

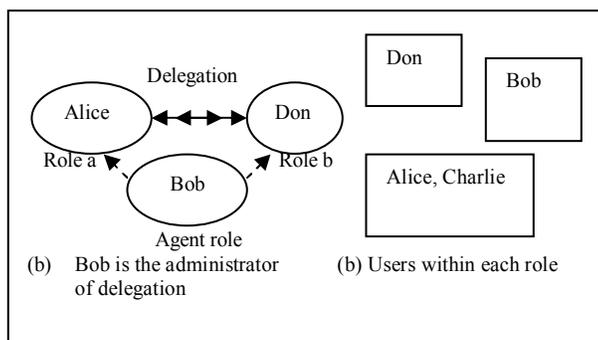


Fig 2. Example of Non- Role Participant Agent Delegation

For example, using the physical example described above, the agent can only hand the key to the delegate member, but cannot (himself/herself) enter the room. In cyberspace, a user who is a member of an agent role can only assign one member of a given role to another role in order for the second to carry out some task on behalf of the first.

• Occurrence of Non-Role Participant Agent Delegation

The non-role participant agent delegation can also occur in a similar ways to that of the role participant agent: dynamic and static delegations. In the static agent, the organization "Security officer" defines an agent role with permissions that give any member of that role the power to administer delegation between different users that belong to different roles.

In the case of dynamic agent-based delegation, the delegating role member dynamically assigns an agent by delegating his role membership to another user who belongs to another role for the purpose of further delegating that role to a third user. The agent in this case will be authorized to administer the delegation. This type of delegation is similar to that of the two-step delegation in that the delegating user dynamically selects a third party "agent" and delegates his role to this third party (agent) for the purpose of administering the delegation of his role to another user that belongs to another role. The agent in this case does not become a full member of the delegating role, and he will not possess the power to operate in that role. For example, in physical space, suppose that Alice is a physician in some department in a hospital. Now suppose that Alice wishes to give access to Bob, who is her assistant, to enter her office in order to complete some task that Alice started and was unable to complete. Alice can accomplish this by giving the key of her office to Charlie who in turn can allow Bob to enter Alice's office. In this scenario, Charlie's role is strictly administrative, and he cannot enter the room himself.

In the next section we will define a model for agent-based role delegation using a dynamic, non-role participant agent delegation. Models for the other manifestations can be similarly developed. Thus, we will discuss them in a lesser level of detail.

B. Model for Agent-based Role Delegation/Dynamic Non-Role Participant Agent (ABRDM-DNRP)

In this delegation model, the delegating role member initiates the delegation process by delegating his role to a third party with the authority to further delegate that role but not to use it.

This agent can be a member of any other role, but with added delegation administration responsibilities.

We first give some assumptions and basic elements that are specific to this model:

The agent role is a one member role, assigned by the delegating user.

The delegation is on behalf of the delegating user. This means that a member of the agent role, who can be a member of another role, is used as a relay between the delegating user and the delegate user.

The delegation addressed by this model is a one-step delegation. This means once the agent administers the delegation, that delegated role cannot be further delegated. Hence, only the original members of the delegating role "or the administrator" can delegate.

To simplify revocation, we assume that any original member in a delegating role can revoke the delegation of any delegate member in that role. We call this type of revocation grant independent.

We also assume that each unit of delegation has a time element associated with it called duration (T). The duration of each delegation is under the control of the delegating agent. Once the assigned time for the delegation expires, the delegation is automatically revoked.

The delegation is monotonic. This means that during delegation, the delegating role member never loses his power in his role, which means that he can override the delegations of the agent.

The following formalizes the above discussion:

Definition 1: The ABRDM-DNRP model has the following components:

AR is an agent role, which is a regular role with added delegation administration responsibility.

$UAA \subseteq U \times R$ is many to many agent member to role assignment relation

$UA = UAO \cup UAD \cup UAA$

$UAA \cap UAD = \emptyset$ Agent and delegate members in the same role are disjoint.

$Users_O(r) = \{U \mid (U, r) \in UAA\}$

Where: UA is the user assignment; UAO is the user assignment of the original members; UAD is the user assignment of the delegate members; and UAA is the assignment of the agent members.

1. Delegation in ABRDM-DNRP

In ABRDM-DNRP, our goal is to provide a controlled agent-based delegation model. We want to be able to control to whom the agent can delegate and on whose behalf. Our model allows the agent to delegate on behalf of the user only. This means that the agent cannot delegate on behalf of anyone else other than the delegating user nor can he/she delegate on behalf of him/herself. This can be achieved using the following relation:

Definition.2: ABRDM-DNRP controls role-role delegation by means of the relation can-delegate $\subseteq R \times AR \times R$

The meaning of $(a, b, \{c\}) \in \text{can-delegate}$ is that a user (say, Alice) who is a member of a role a can delegate her role membership to any another user (say, Bob) who is an original member of another role b, who can further delegate that role to another user (say Charlie) who is a member of role c. For example, if $Alice \in User_O(a)$, $Bob \in User_O(b)$, and $Charlie \in User_O(c)$, then Alice can delegate a to Bob, who can further delegate a to Charlie, so thereby $(Charlie, a) \in$

UAD. Bob, in this case, acts as an agent for Alice. See Figure 3.

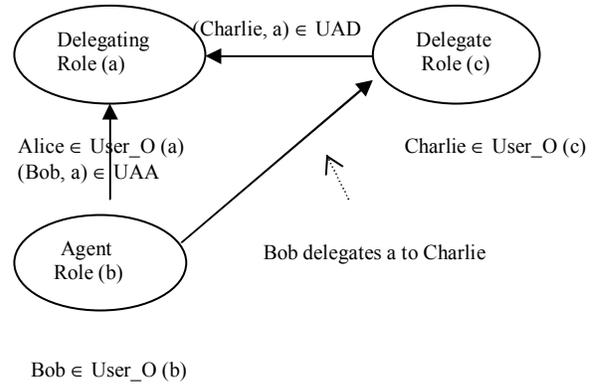


Fig 3. Example of Agent Based Delegation-Dynamic-Non-Role Participant Agent

2. Revocation in ABRDM-DNRP

So far we have described how users in a delegating role can delegate their permissions to other users who are in other roles and we have demonstrated how we can control this process using the can-delegate relation. However, as often happens in real life, we may want to revoke rights. For example, suppose that the department chairperson in university goes away and his role is delegated to another physician. That delegate physician then becomes the acting chairperson, and thus obtains the permissions assigned to the chairperson. Subsequently, when the department chairperson returns, the delegated permissions need to be removed from the delegate physician. With the agent-based delegation, the delegation is temporary [BS2000]. Moreover, the agent does not play a role in the revocation process. Therefore, in this model, revocation is handled the same way as in RBDM0 [BS2000].

ABRDM-DNPA deals with the issue of revocation in two ways: by using timeouts and by allowing any original member of the delegating role to revoke the membership of any delegate member in that role. We call the latest grant-independent revocation. Both of these approaches are discussed in detail in RBDM1 [BS2004], and thus will not be addressed any further in this section.

II. AGENT-BASED ROLE DELEGATION MODEL IN HIERARCHICAL ROLES (ARBDMH)

So far, our discussion of the agent-based role delegation has been within the context of the flat role relation. We now turn our attention to agent-based role delegation in the cases where the relation between the roles is hierarchical.

In hierarchical roles, senior roles inherit the permissions of roles that are junior to them. When we extend our model to capture the role-to-role delegation using hierarchical roles, this

adds more complexity to our flat model. Here, we have to deal with different kinds of delegations, some of which are not very useful and some which carry more risk than others do.

The motivation behind delegation using hierarchical roles was addressed in RBDM1 [2004], and the motivation behind agent-based delegation was addressed in an earlier section of this paper. Therefore, this section will focus on the impact of the hierarchical role relation on the agent-based delegation. More specifically, we are going to develop an agent-based delegation model in hierarchical roles, we call this model ARBDMH, by taking the agent-based delegation model in flat role, addressed in the previous section, and extending it by including hierarchical roles. Then we analyze the impact that these changes have on agent-based delegation.

First, we identify some key elements that apply to agent-based role delegation in hierarchical roles.

Since this model is an extension to the delegation model using flat roles discussed in the previous section, we add the following assumptions and basic elements that apply specifically to the delegation model using hierarchical roles:

- Delegation can only be either downwards or cross. Upwards is useless because senior roles inherit all the permission of their junior roles.
- Due to the inheritance nature of role hierarchies, the agent is limited to a certain range of delegation.
- In hierarchical roles, a member of a role that is senior to the agent role is also an agent.
- The addition of role hierarchy introduces a new notion for a user membership in a role (explicit and implicit memberships). The explicit role membership grants a user the authority to use the permissions of that role because of his/her direct membership to that role. The implicit role membership, on the other hand, grants a user the authority to use the permissions of that role because of that user's membership of a role that is senior to the given role.
- Combining the two new types of role memberships with the original two types (original memberships and delegate memberships) produces 4 different combinations of user memberships in each role at any given moment. These combinations are: original/explicit, original /implicit, delegate/explicit, and delegate/implicit
- Our model allows only members of original/explicit and original/implicit roles to serve as agents.
- Revocation issues become more complicated when we deal with hierarchical roles. This is because of the involvement of many different roles.

A. Delegation in ARBDMH

In large enterprise-wide systems, the number of roles can be in the hundreds and the number of user can be the tens of thousands. In large systems, administering delegation of roles where the relation between these roles is hierarchical can be a

formidable task, and if it is not carefully controlled it can lead to chaos.

In ARBDMH, we are going to borrow the concept of user-role assignment discussed in the role-based administrative model, by Sandhu [5], to address the agent-based role delegation between users in hierarchical roles.

URA97 is concerned with administration of the user-assignment relation UA, which relates users to roles. Authorization to modify this relation is controlled by administrative roles. Thus, members of the administrative roles in Figure 4 (b) are authorized to modify the membership in the roles of Figure 4 (a). Assignment of users to administrative roles is outside the scope of URA97, and is assumed to be done by the chief security officer. Therefore, it is outside the scope of this model also.

ARBDMH is concerned with the administration of role delegation between users who are members of different roles. This is also an administration of user-assignment relation UA. The main difference here is that in ARBDMH the delegation is discretionary.

The assignment of users to the administrative roles (in our model these administrative roles are called agents) takes two forms: dynamic and static assignment.

We use the hierarchies of Figures 4(a) and 4(b) in our running example through this section of the paper to illustrate how our model controls the range of authority for the agent. Figure 4(a) shows the regular roles in the engineering department. There is a junior-most role E to which all employees belong. The engineering department has a junior-most role ED and a senior-most role DIR. In between there are roles for two projects within the department, project 1 on the left and project 2 on the right. Each project has a senior-most project lead role (PL1 and PL2), a junior-most engineer role (E1, E2), and in between two incomparable roles, production engineer (PE1 and PE2). There is also a quality engineer roles (QE1 and QE2). Figure 4(b) shows the agent role hierarchy with the senior delegating agent (SDA) role at the top, two project delegating agents (PDA1 and PDA2), and a department delegating agent (DDA) role.

In ARBDMH, our goal is to impose restrictions on which users can be delegated to and by which agent. The notion of a prerequisite condition is a key part of ARBDMH.

Definition 3: In the ARBDMH model, prerequisite condition CR is a Boolean expression using usual "&" (and) and "|" (or) operations on the term of cr. cr is evaluated for a user u to be an original member in a role by interpreting x to be true if $u \in \text{User-OE}(r) \vee u \in \text{User-OI}(r)$, and $\neg x$ is true if $u \notin \text{User-OE}(r) \vee u \notin \text{User-OI}(r)$

This means that x denotes an original membership (explicit or implicit) and $\neg x$ denotes absence of the original membership.

Definition 4: The role-role delegation is authorized in ARBDMH by the following relation:

$$\text{Can-delegate} \subseteq \text{AR} \times \text{CR} \times 2\text{R}$$

The meaning of $(x, cr, z) \in \text{can-delegate}$ is that a user who is a member of role x (or any role senior to x), and whose membership in x satisfies the prerequisite cr , can delegate to any another user who is a member of role y , where $(y \leq x)$.

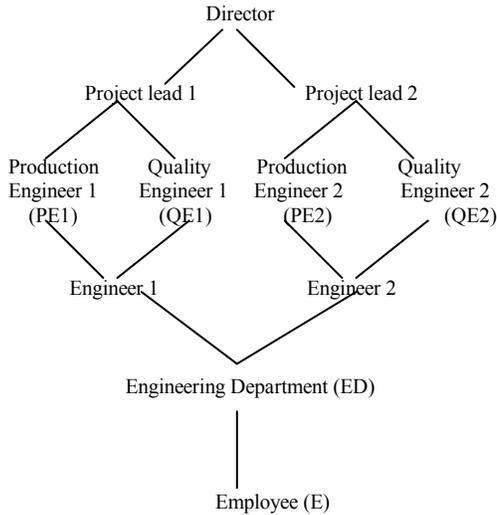


Fig. 4(a). An Example Role Hierarchy

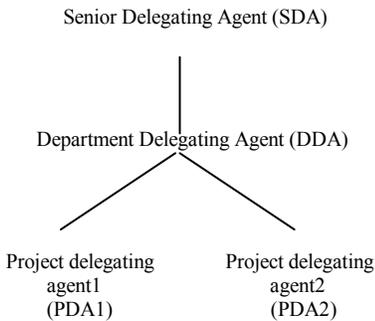


Fig. 4(b). An Example Agent Role Hierarchy

Table 2 illustrates the can-delegate relations. Role sets are specified in ARBDMH by the following range notion:

$$[x, y] = \{r \in R \mid x \geq r \wedge r \geq y\} \quad [x, y) = \{r \in R \mid x \geq r \wedge r > y\}$$

$$(x, y] = \{r \in R \mid x > r \wedge r \geq y\} \quad (x, y) = \{r \in R \mid x > r \wedge r > y\}$$

By table 2, PDA1 can delegate the roles E1, PE1, and QE1 to users in these roles, provided that delegated users have a membership in ED as a minimum. Similarly, PDA2 can delegate the roles E2, PE2, and QE2. DDA can delegate the roles of ED to PL1 to users of these roles provided that the user is a member of ED, as a minimum, and does not have membership in PL2, and similarly for PL2 with respect to PL1.

Table 2

Example of Can-Delegate

Agent Role	Prerequisite Condition	Delegation Range
PDA1	ED	[E1, PL1)
PDA2	ED	[E2, PL2)
DDA	$ED \wedge \neg PL1$	[PL2, PL2]
DDA	$ED \wedge \neg PL2$	[PL1, PL1]

B. Revocation in ARBDMH

In agent-based role delegation model, delegation is temporary. Therefore, it is very important to control the way revocation is handled in ARBDMH. A notable aspect of revocation in ARBDMH is that revocation is independent of assignment. This means that revocation can occur without any involvement by the delegating agent. In ARBDMH, there are two ways by which delegation can be revoked:

- Revocation using timeout
- Human revocation

In the revocation using timeout approach, a duration constraint is attached to each delegation relation so that when the assigned time expired, the delegation is also expired.

Duration-restriction a self-triggered process that ensures the automatic revocation of the role membership. It is extremely useful when the attached duration is a small time period. It eliminates the overhead that can results form the administration of manually revoking a delegation.

Revocation using timeout by itself is not enough to ensure security; and time period must be set with great deal of care, to avoid any oversight, or under setting the time period.

The other approach to doing revocation of delegation in ARBDMH is by humans. We consider two types of humans to implement revocation of delegated roles. Security officer, and the original users in the delegating role. The earlier, gives the power to the security officer to revoke the membership of any delegate user. This is due to a bad behavior by the delegate user in the delegated role, or because of the lack of need by the organization for the delegate user to remain a member in the delegated role.

The later, allows any original member of the delegating role to revoke the membership of a delegate user. This gives the power to the original members to protect the role from the temporary delegate members, If the delegate member behaves badly, any original member can revoke him immediately, which will minimize the damage before even the time out. However, It raises the possibility of conflicts between the original members if someone else other than the sponsoring original member revokes the delegate member.

The following defines the can-revoke relation in ARBDMH.

Definition 5: The role-role revocation is authorized in ARBDMH using the following relation,

$$\text{Can-Revoke} \subseteq R \times 2R$$

The meaning of can revoke $(x, y) \in \text{can-revoke}$ is that the delegating member of role x (explicit or implicit) can revoke the membership of the delegate member y or any subsets of y in the role x .

III. CONCLUSION

In this paper, we have addressed the agent-based role delegation, which is one of delegation characteristics described in the literature by Barka and Sandhu [BS2000]. We have described a systematic approach in which an agent-based delegation can be implemented. We have identified two manifestations, role-participant agent and non-role participant agent, to delegation using agent-based role delegation. Furthermore, we identified two additional modes in which these delegation can occur: static and dynamic. We used the dynamic non-role participant agent, manifestation to develop our model for agent-based role delegation. Models to describe the other manifestations can be similarly developed, thus were briefly mentioned in this paper.

REFERENCES

- [1] Martin Abadi, Michael Burrows, Butler Lampson and Gordon Plotkin. A calculus for Access Control in Distributed Systems. *ACM Transactions on Programming Languages and Systems*, Vol. 15, No 4, September 1993, pages 706-734.
- [2] David Ferriolo and Richard Kuhn. Role-based access controls. In *Proceedings of 15th NIST-NCSC National Computer Security Conference*, pages 554-563, Baltimore, MD, October 13-16 1992.
- [3] Morrie Gasser, Ellen McDermott. An Architecture for practical Delegation in a Distributed System. 1990 IEEE Computer Society Symposium on Research in Security and Privacy. Oakland, CA. May 7-9, 1990.
- [4] B.W. Lampson, Protection. 5th Princeton Symposium on information science and systems. Pages 437-443.
- [5] Ravi Sandhu and Venkata Bhamidipati. Role-based administration of user-role assignment: The UR97 model and its Oracle implementation. In *Proceedings of IFIP WG11.3 Workshop on Data Security*. August, 1997.
- [6] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38-47, February 1996.
- [7] Ezedin Barka and Ravi Sandhu. A Role-based Delegation Model and Some Extensions. *Proceedings of 23rd National Information Systems Security Conference*, Pages 101-114, Baltimore, Oct. 16-19, 2000
- [8] Ezedin Barka and Ravi Sandhu. Framework for Role-Based Delegation Models. In *Proceedings of 16th Annual Computer Security Application Conference*, New Orleans, LA, December 11-15 2000
- [9] Ezedin Barka and Ravi Sandhu. Role-Based Delegation Model/ Hierarchical Roles (RBDM1). In *Proceedings of 20th Annual Computer Security Application Conference*, Tucson, AZ, December 6-10 2004