Adaptive Access Control Modes Enforcement in Organizations*

Sérgio Guerreiro[1], André Vasconcelos[1,2], and José Tribolet[1,2]

[1] CODE - Center for Organizational Design & Engineering, INOV, Rua Alves Redol 9, Lisbon, Portugal
[2] Department of Information Systems and Computer Science, Instituto Superior Técnico, Technical University of Lisbon, Portugal
sergio.guerreiro@ist.utl.pt, andre.vasconcelos@dei.ist.utl.pt, jose.tribolet@inesc.pt

Abstract. Granting the correct access between the agents and the artifacts is nowadays in the organizations agendas. The risk of allowing unauthorized accesses to critical information requires new solutions that are capable of dealing with a holistic perspective. Adaptive OACM refers to the capability of enforcing fine-grained access policies to business processes, services and information systems whenever facing changes, for instance, governance policies. This paper proposes an OACM ontology based in the RBAC, UUID, Rules and architectural model concepts. For exemplification purposes we instantiate the concepts of the ontology to an approval expense problem.

Keywords: ACM, RBAC, Artifacts, Organization, Workflow, Services, Informational entities.

1 Problem statement

The access control modes (ACM) that are necessary to authorize a fine-grained access to organizational artifacts bounds an organizational wide problem. Moreover, if the artifacts are located in different architectural layers of an organization, or cross organizations, then an additional effort is needed. Organizational ACM (OACM) is thus defined as the structural aspects for granting or revoking the correct access between the agents and the artifacts. Typically, the ACM strategies are applied to silos inside the organization [5][9]. For each silo, a set of well-known requirements is used, *e.g.:* applications authorization, operating systems authorization or database authorization. However those approaches are not suitable for an organization-wide perspective. With this work we seek for a complete fine-grained rastreability between the agents (either Human or machines), the artifacts, their actions and the orchestration between the actions. An application example for this endeavour is the cloud computing environment [19][20] where the access to the artifacts in the cloud must consider the interoperability between the different Persons and systems working in a integrated manner.

* This work was partially supported by the Fundação para a Ciência e a Tecnologia (SFRH / BD/ 43252 / 2008).

J.E. Quintela Varajão et al. (Eds.): CENTERIS 2010, Part II, CCIS 110, pp. 283–294, 2010.

From other point of view, the adaptability quality of an information system is broadly referenced in the foundations of Software Engineering, *e.g.*: the recommended practices for software requirements IEEE 830-1998 [17] and the software architecture [21]. The software adaptability is identified, among others, as extra-functional properties of a system that should be included in its design. The adaptability quality in Software Engineering aims in minimizing the software deterioration due to change [18]. Also in the normalized systems theory presented by Mannaert *et al.* [13] a strong focus is given to the adaptability of a system: the postulate 1 defines that *"a design pattern needs to be stable with respect to anticipated changes"*. We define the OACM adaptability quality as the capability to enforce new or modified access policies the organizational artifacts, in real time, with the minimal effort. To enforce the OACM we consider the artifacts from the following architectures: business processes, services and informational entities.

Therefore, with this paper we seek a solution to the adaptive fine-grained ACM enforcement in an organization by proposing a complete ontological model that includes adaptability concerns, in order to allow a fully configurable access authorization.

At this point its also relevant to state that the U.S. Department of Homeland Security [14] defines the following objectives that are strongly related with the ACM body of knowledge: *(i)* assigned in the strategic goal #5:*"Integrate DHS Policy, Planning, and Operations Coordination. We will strengthen and unify strategic and policy direction through improved strategic planning and assessment. We will ensure that these efforts are integrated with and informed by the Department's operations coordination and planning efforts. We will create and enhance a DHS operations coordination capability to plan for and coordinate non-routine, cross-cutting operations that require multi-Component activities"*, and *(ii)* assigned in the strategic goal #3:*"Improve Cyber Security. We will reduce our vulnerabilities to cyber system threats before they can be exploited to damage the Nation's critical infrastructures and ensure that such disruptions of cyberspace are infrequent, of minimal duration, manageable, and cause the least damage possible"*. These definitions reinforce the need to further investigate this problem, offering solutions that consider the overall complexity inside one organization.

This paper is the result of an action research methodology effort supported by an OACM ontology simulator, which is implemented in OO and relational database schema. It is exemplified with a classical approval expenses problem in an organization. This methodology allowed the experimentation of problem resolution strategies, as well as induced an iterative, detailed and coherent implementation of each ontology concept. The presented example is specifically concerned with the OACM issues rather than the architectural rastreability of the artifacts in the different layers. However, for OACM demonstration purposes of the fine-grained access control to organizational artifacts we defined a static relationship between them.

The rest of the paper is organized as follow. Section 2 presents the related work with the role based access control (RBAC) ACM and the existing efforts to apply it to organizations. Section 3 proposes an OACM conceptual meta-model and defines the main concepts. Section 4 exemplifies the OACM applied to a simplified approval expenses problem. Finally, Section 5 concludes and points to future work.

2 Related Work

2.1 Role Based Access Control

The standard NIST98 [5][9] models the concepts for symmetric role-based access control (RBAC) ACM to be used between the users, roles, permissions and constraints. It represents an evolution from the Discretionary Access Control (DAC), Mandatory Access Control (MAC) and other policies due to less provisioning effort needed [10]. The users are directly assigned to a role, each role has a set of associated permissions and changing the permissions affects the users associated with each role. Some well-known constraints are: separation of duties (SoD), conflict of interest (CoI), delegation of duties (DoD), binding of duty (BoD), history-based separation of duties (HsD) to newly identified constraints in the social networks such as the context constraints [4]. This model is applicable to organizational silos; however it is limited to only one kind of organizational artifact, at a single time. The NIST98 is broadly used in single architectural layers such as applications or databases [12], however the enforcement in WfMS is still an unsolved challenge [6]. Bertino *et al.* in [2][3] proposes the combination of static, dynamic and hybrid constraints to split the enforcement of RBAC in WfMS: the static constraints are processed offline and the dynamic constraints requires an execution engine to monitor the workflow sessions. Wolter *et al.* in [11] defines a set of workflow primitives and then experiments the concept of notations to express the associated constraints, however the concept of a transaction (rollback and commit concern) is not addressed by this proposal.

2.2 Organizational RBAC

Since there are no separation between the organizational and the system roles in the RBAC approach then it is not satisfactory to control the artifacts spread through the different architectures of an organization. This is true only in organizations where each Person has the same system and hierarchical role [10].

Park *et al.* in [8] proposes to unify ACM by separating the organizational and the system roles. The OR-SRA maps them, in real-time, using the concepts of constraints, roles, permissions, sessions and hierarchy. The OR-SRA is mapped accordingly with a predefined set of relations, requiring an extra effort of role engineering [1].

An alternative solution is proposed by Myong *et al.* in [7] under the scope of inter-enterprises business processes execution. Each enterprise has its own RBAC and the role domain of each one is passed through the communications. The authors argue that their approach *(i)* separates the application-level from the organization-level, *(ii)* achieve a fine-grained control and *(iii)* supports dynamic constraints. However, this proposal does not encompass the time and versioning concern, neither relates disparate architectural layers.

Zhixiong Zhang PhD thesis's in [15] extends the NIST98 standard with the concepts of Organization and Asset, justified by the incapacity of the model for the cross-organization ACM. The aim is to create a ROBAC between organizations that possess different assets. The roles are mapped with the assets and each asset has its own permission set, but the assets are not further detailed neither the relation between the assets. It also lacks in the assumption that the organizations have comparable roles. To enforce it uses a manifold that implements a virtual organization which refers to the involved assets. By security concerns, in the end it is deleted.

3 Proposed Solution

Fig. 1 depicts our proposed OACM concepts. The `users` are the ones that take actions in the organization, they might be either Persons or machines. The `model` is the core concept representing the architecture layers considered to the access control: *(i)* Workflow, is a set of orchestration steps, encompassing agents and actions; *(ii)* Services, are the actions performed; and *(iii)* Information entities, represents the entities that are computed by the actions. Each model is implemented by a set of `Fieldmodel` which represents the detail of each of the model. Further detail for each `Fieldmodel` can be expressed by the `Property` concept. Each `Fieldmodel` establish the `permission` required to execute it. Each `Fieldmodel` also requires the previous definition of a `role`.

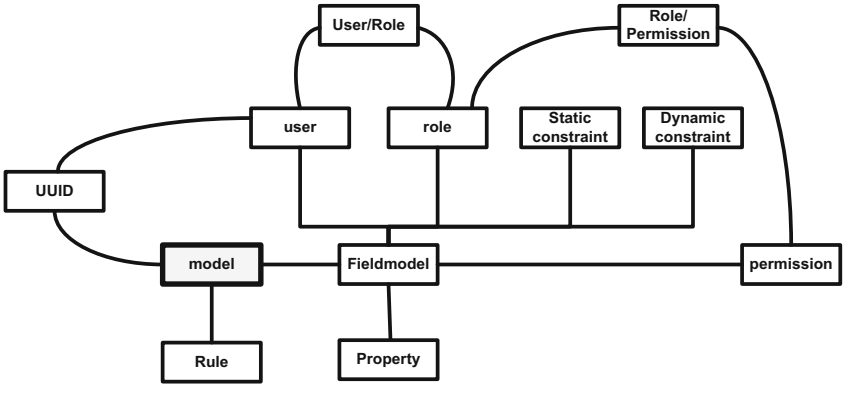

Fig. 1. OACM conceptual meta-model

Following the RBAC principles, an `user` can only execute a `Fieldmodel` if and only if *(i)* his `role` is assessed successfully by the `User/Role` mapping, *(ii)* his `role` is checked in the `Role/Permission` mapping and *(iii)* none of the `static constraint` and `dynamic constraint` are triggered. For exemplification purposes, the `static` and the `dynamic constraint` are implementing by separation of duties (SoD) policy.

Each `model` defines their set of `users` and `roles`, resulting in the creation of different identifiers for the same `user` or `role` through the 3 distinct `models`. To solve this issue, the `UUID` concept defines a transversal identification of `users` through the different `models` of the organizational.

Fig. 2 distinguishes the concepts of `model` and `session`, the first is composed by `Fieldmodel`, `roles` and `static constraint` and the second is an instantiation of the `model` by a `user` and by `dynamic constraint`. A transformation between `Role` and `User` is also required in the session.

The enforcement of the RBAC in the sessions represents the local access control, *e.g.:* each user invocation of a task within a workflow is always checked to verify if the role is correct, the permission allows and no constraint is triggered. However, a

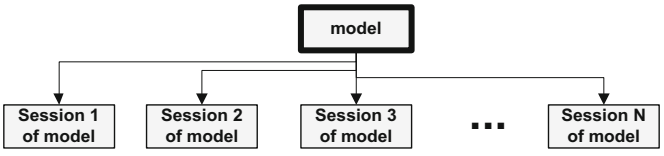

Fig. 2. Architecture of workflow, services and informational entities: `Model` and `Session`

global access control that is concerned with the relationship between the `Session` is also required, *e.g.:* the access to the workflow tasks is granted but the access to the information entities is not granted due to personal relationships of the user. To solve this issue we propose the concept of the `Rule` execution between different `Models`. The `rules` have *adhoc* definition.

Therefore, the expected results from the OACM are *(i)* the access check to the artifacts of each `model` and *(ii)* the access check between the artifacts of different `models`. The output produced corresponds to a grant or to a revoke.

3.1 Advantages of the Proposed Solution

The following advantages are identified with the proposed OACM: *(i)* not intrusive to the organizational artifacts while it establishes a virtual architecture that intercepts the execution of the sessions; *(ii)* `models` are tailor-made and with least privileges enforcement; *(iii)* `models` representation could be independent from the sessions IS architecture representation used by the organization; *(iv)* a virtual architecture only exists when needed, it is deleted in the end of the execution of the sessions, improving the security; *(v)* a virtual architecture offers an adaptability mechanism to be used by the organization and *(vi)* it is possible to refining the ACM policy in each `model`.

3.2 Primitives for the OACM Proposal

From the above concepts we summarize them as a set of definitions to be used by the OACM computational simulation. These set of definitions allow a better understanding of the related concepts as well as facilitate the development process of the simulator.

Definition 1: A `model` is a static representation of a workflow, or services, or informational entities, which are contained in an organizational environment, encompassing the `FieldModels`, the `roles` and the `staticConstraints`. The instantiation of a `model` is a `session`. A `session` includes the same information as the `model` except concerning the *(i)* `users` and their transformation process to `Roles` and the *(ii)* `dynamicConstraints`.

Definition 2: Each Person or machine, contained in and organizational environment have unique identities. The unique identities apply to the `role`, `permission`, `staticConstraint`, `dynamicConstraint` and `user`. The RBAC approach is used to enforce each identity to his `Role` and `Permission`.

Definition 3: Consider that a `model` is represented by a graph *G* composed by artifacts *A* and relationships *R*. *UR(A=user)* represents the user mapping to a role.

RP(A=permission) represents the permission mapping to the correspondingly role, and that *[]* represents the computation of a constraint. Then, the check access assessment, for one specific model, is computed by the eq. **(1)**, producing a result of *true* or *false*.

$$\begin{aligned}&\text{CheckAccess}() = \\ &G_{model}(A=role\ ,\ R) \cap G_{session}(A=UR(user)\ ,\ R)\ \cap \\ &G_{model}(A=RP(permission)\ ,\ R) \cap \\ &G_{model}(A=role[staticConstraint],R) \cap G_{session}(\ A=UR(user[dynamicConstraint]),R)\end{aligned} \tag{1}$$

Definition 4: The check access assessment between two different `models` is performed by *adhoc* `rules`. The rules are composed by a set of operators and predefined operations. The operators are taken from the `Model` and `session`. The structure of the *adhoc* rules is virtual and is deleted in the end of the usage.

Definition 5: The adaptability of the OACM is offered by: *(i)* the RBAC mapping, *(ii)* the `staticconstraint` and `dynamicconstraint`, *(iii)* the `uuid` mapping, *(iv)* the `fieldmodel` defined in each `model`, *(v)* the `property` defined in each `fieldmodel`, *(vi)* mapping functions such as *UR()* and *RP()* presented in definition 3 and *(vii) adhoc* `rule` definition.

Definition 6: each OACM is only valid in a limited timeframe.

4 Expenses Approval Example

A simplified expenses approval scenario is used to demonstrate the applicability of the proposed OACM. Fig. 3 summarizes the set of activities involved in the expense approval. If the employee expense's are less or equal than 500€ than the manager is able to approve from a predefined budget, however for expenses greater than 500€ a escalate procedure to the director is required. At any time, the expense requests might be rejected or cancelled. The result of enforcing the equation (1) in the models of workflow, services and information entities with a predefined access structure is a granting or a revoking to the desired access. The predefined access structure is presented in subsections 4.1 and 4.2, the models are presented in subsection 4.3, and the global enforcement in subsection 4.4.

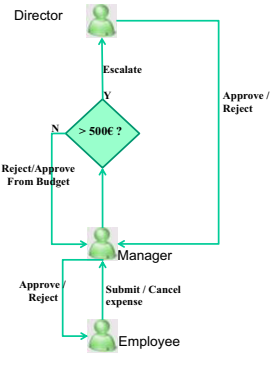

Fig. 3. Approval expense interactions including 3 Persons: employee, manager and director

4.1 Unique Identities

Regarding the definition 2, and following an approach similar to [16], Table 1 presents the universal mapping for each Person that is involved in the expense approval problem. Each Person has a different user for each model, which must be maintained separately. The UUID column allows the definition of unique identifiers to be used when designing the rules between different models.

Table 1. `UUID` **-** Universal access table

Person	UUID	Workflow User	Application User	Information user
António	0001	Actor1	A001	E_APP
João	0002	Actor2	J002	M_APP
Manuel	0003	Actor3	M003	D_APP

4.2 RBAC for Local Access Control

Also regarding the definition 2, each model have a specific RBAC definition, where *(i)* the workflow RBAC is defined by Table 2 and Table 3, *(ii)* the services RBAC is defined by Table 4 and Table 5 and *(iii)* the informational entities is defined by Table 6 and Table 7. Each Person has a User, with a set of related Roles for accessing each Model, and correspondingly each Role is also related with a set of Permissions for that Model. It is possible to manage (*e.g.:* add, remove or change) the Users associated with a Role and it is also possible to manage the Permissions associated with a Role.

Table 2. `UR` relationship for workflow

Person	User Workflow	Role Workflow
Antonio	Actor1	Employee
Joao	Actor2	Director
Manuel	Actor3	Manager

Table 3. `RP` relationship for workflow

Role Workflow	Task Permission
Director	Task3
Employee	Task1
Manager	Task4
Manager	Task2

Table 4. `UR` relationship for services

Person	User Service	Role Service
Antonio	A001	EmployeeService
Joao	J002	ManagerService
Manuel	M003	DirectorService

Table 5. `RP` relationship for services

Role Service	Service Permission
DirectorService	ApproveExpense()
DirectorService	RejectExpense()
DirectorService	Login()
DirectorService	Logout()
EmployeeService	ExpenseSubmission()
EmployeeService	ExpenseCancellation()
EmployeeService	Login()
EmployeeService	Logout()
ManagerService	RejectFromBudget()
ManagerService	Logout()
ManagerService	Login()
ManagerService	ExpenseEscalate()
ManagerService	ApproveFromBudget()

Table 6. UR relationship for information entities

Person	User Information Entity	Role Information Entity
Antonio	E_APP	EI_Employee
Joao	M_APP	EI_Manager
Manuel	D_APP	EI_Director

Table 7. RP relationship for information entities

Role Information Entity	Operation Permission
EI_Director	UPDATE
EI_Director	READ
EI_Director	DELETE
EI_Employee	READ
EI_Employee	CREATE
EI_Employee	DELETE
EI_Manager	DELETE
EI_Manager	CREATE
EI_Manager	UPDATE
EI_Manager	READ

4.3 Model Definition

Regarding the definition 1 for models and sessions, Table 8, Table 9 and Table 11 define the three separate models, presented in a tabular form. Firstly, Table 8 presents the sequence of tasks that are necessary to define the workflow of expenses greater than 500€, using the notation proposed in [13]. A static constraint of SoD is defined, meaning that is not allowed for the Director or Manager to approve their own expenses.

Table 8. Model definition for the approval expense workflow

Start State	End State	Task name	Role WF	Static constraint
Creation	Submitted	Task 1, Submiting	Employee	SoD
Submitted	Account available	Task 2, checking account	Manager	SoD
Account available	Approved	Task 3, approving	Director	SoD
Approved	Reimbursed	Task 4, reimbursing	Manager	SoD

Table 9. Model definition for the approval expense services

Service	Role Service	Static Constraint
ApproveExpense()	DirectorService	
Login()	DirectorService	
Logout()	DirectorService	
RejectExpense()	DirectorService	
ExpenseCancellation()	EmployeeService	
ExpenseSubmission()	EmployeeService	
Login()	EmployeeService	
Logout()	EmployeeService	
ApproveFromBudget()	ManagerService	
ExpenseEscalate()	ManagerService	
Login()	ManagerService	
Logout()	ManagerService	
RejectFromBudget()	ManagerService	

Table 10. Relation between the Services and the informational entities

service \ Information entities	Document	authorization	account	budget
Login()		R		
ExpenseSubmission()	C			
ExpenseCancellation()	RD			
Logout()		R		
Login()		R		
ApproveFrombudget()			CR	CR
RejectFromBudget()			RD	RD
ExpenseEscalate()	RU			
Logout()		R		
Login()		R		
ApproveExpense()			RU	
RejectExpense()			RD	
Logout()		R		

Table 11. Model definition for the approval expense information entities

Information entity	Operation	Role IE	Static constraint
Account	DELETE	EI_Director	
Account	READ	EI_Director	
Account	UPDATE	EI_Director	
Authorization	READ	EI_Director	
Document	READ	EI_Director	
Authorization	READ	EI_Employee	
Document	CREATE	EI_Employee	
Document	DELETE	EI_Employee	
Document	READ	EI_Employee	
Account	CREATE	EI_Manager	
Account	READ	EI_Manager	
Acount	DELETE	EI_Manager	
Authorization	READ	EI_Manager	
Budget	CREATE	EI_Manager	
Budget	DELETE	EI_Manager	
Budget	READ	EI_Manager	
Document	READ	EI_Manager	
Document	UPDATE	EI_Manager	

Secondly, Table 9 presents which services are used by each Role. The service invocation sequence is depicted in Fig. 3. None static constraint is used.

Thirdly, Table 11 specifies the information entities that are used in expense approval. Each information entity is involved in one or more operation. The informational entities that are used in each service are presented in the CRUD matrix by Table 10. None static constraint is considered.

4.4 Model Execution and Global Access Control by Rules

Regarding the definition 3, for each model from subsection 4.3, the check access assessment is performed, using the equation (1). When a row of the table is delivered then it means that the access is granted for that Fieldmodel. As expected, three different areas are identified (by thicker borders) in Table 12: the workflow, the services and information entities. The user identifies the running sessions in each area. The Persons that execute the sessions are always one of the three involved: Antonio, Joao or Manuel. Any dynamic constraint must be also considered in this step. To guarantee security when all sessions stop then this access table must be deleted.

Table 12 also presents the result of applying one rule presented by the *adhoc* equation (2) to the local check access assessment table. The result is represented by shadowed cells, it expresses the dependency between the workflow model and the informational entities layers. In this case, a session performed by Antonio in submitting the expense, automatically grants the authorization to the UPDATE account by Manuel. Therefore, the rule allow the cross model access control, which is essential to a holistic perspective of the organization.

CheckGlobalAccess() =
`/* automatic approval for Antonio expenses*/` (2)
IF (Antonio submits expense in task1) THEN Manuel UPDATE Account;

Table 12. Execution sessions and global access control

Person	All User	All Role	workflow task	service	information entities	
Antonio	Actor1	Employee	Submiting - task1			
Manuel	Actor3	Manager	CheckingAccount-task2			
Joao	Actor2	Director	Approving - task 3			
Manuel	Actor3	Manager	Reimbursing - task 4			
Antonio	A001	EmployeeService		Login()		
Antonio	A001	EmployeeService		ExpenseSubmission()		
Antonio	A001	EmployeeService		Logout()		
Joao	J002	ManagerService		Login()		
Joao	J002	ManagerService		ApproveFromBudget()		
Joao	J002	ManagerService		ExpenseEscalate()		
Joao	J002	ManagerService		Logout()		
Manuel	M003	DirectorService		Login()		
Manuel	M003	DirectorService		ApproveExpense()		
Manuel	M003	DirectorService		Logout()		
Antonio	A001	EmployeeService		ExpenseCancellation()		
Joao	J002	ManagerService		RejectFromBudget()		
Manuel	M003	DirectorService		RejectExpense()		
Antonio	E_APP	EI_Employee			Document	CREATE
Antonio	E_APP	EI_Employee			Authorization	READ
Antonio	E_APP	EI_Employee			Document	READ
Antonio	E_APP	EI_Employee			Document	DELETE
Joao	M_APP	EI_Manager			Account	CREATE
Joao	M_APP	EI_Manager			Budget	CREATE
Joao	M_APP	EI_Manager			Account	READ
Joao	M_APP	EI_Manager			Budget	READ
Joao	M_APP	EI_Manager			Document	READ
Joao	M_APP	EI_Manager			Authorization	READ
Joao	M_APP	EI_Manager			Document	UPDATE
Joao	M_APP	EI_Manager			Acount	DELETE
Joao	M_APP	EI_Manager			Budget	DELETE
Manuel	D_APP	EI_Director			Authorization	READ
Manuel	D_APP	EI_Director			Document	READ
Manuel	D_APP	EI_Director			Account	READ
Manuel	D_APP	EI_Director			Account	UPDATE
Manuel	D_APP	EI_Director			Account	DELETE

5 Conclusions and Future Work

This paper proposes ontology for controlling the access to organizational artifacts (OACM). The ontology is exemplified with a simplified expense approval process.
As referred by the definition 5 for the general case, and particularly for this example, the adaptive OACM counterparts are: RBAC and UUID configuration, model definition and constraints and the rules definition between models. These counterparts allow an organization to control his artifacts accesses either by a local or a global perspective.

The OACM does not enclose all the security protocols needed in one organization but it is rather a structural mechanism that allows the fine-grained control access using a non-intrusive approach. Fine-grained is a continuous process of artifacts authorization running on a real-time basis. Furthermore, the enforcement result represents the OACM observability and the adaptive OACM counterpart represents the actuation. Hence, with OACM we obtain a security governance implementation capable of controlling cyber security or coordination security policies, applied to one organization.

We identify the following issues to be further researched as future work: *(i)* a comprehensive implementation of the meta-model in a real case study, the artifacts access information might be collected using interceptors and the grant/revoke operation might be implemented by remote controlling the execution of the interceptors, *(ii)* definition of a complete language set for the rules implementation and *(iii)* further develop the definition 6, integrating the time concerns in the equation (1) to achieve an history based OACM.

References

1. Atluri, V.: Panel on role engineering. In: SACMAT 2008: Proceedings of the 13th ACM Symposium on Access Control Models and Technologies, New York, NY, USA, pp. 61–62 (2008)
2. Bertino, E., Ferrari, E., Atluri, V.: The specification and enforcement of authorization constraints in workflow management systems. ACM Trans. Inf. Syst. Secur. 2(1), 65–104 (1999)
3. Bertino, E., Ferrari, E., Atluri, V.: A flexible model supporting the specification and enforcement of role-based authorization in workflow management systems. In: RBAC 1997: Proceedings of the Second ACM Workshop on Role-based Access Control, New York, NY, USA, pp. 1–12 (1997)
4. Carminati, F.E., Perego, A.: Enforcing access control in web-based social networks. ACM Trans. Inf. Syst. Secur. 13(1), 1–38 (2009)
5. Ferraiolo, D., Sandhu, R., Gavrila, S., Kuhn, R., Chandramouli, R.: Proposed nist standard for role-based access control. ACM Trans. Inf. Syst. Secur. 4(3), 224–274 (2001)
6. Hung, P., Karlapalem, K.: A secure workflow model. In: ACSW Frontiers 2003: Proceedings of the Australasian Information Security Workshop Conference on ACSW Frontiers 2003, pp. 33–41. Australian Computer Society, Inc., Darlinghurst (2003)
7. Kang, M., Park, J., Froscher, J.: Access control mechanisms for inter-organizational workflow. In: SACMAT 2001: Proceedings of the Sixth ACM Symposium on Access Control Models and Technologies, New York, NY, USA, pp. 66–74 (2001)
8. Park, J., Costello, K., Neven, T., Diosomito, J.: A composite rbac approach for large, complex organizations. In: SACMAT 2004: Proceedings of the Ninth ACM Symposium on Access Control Models and Technologies, New York, NY, USA, pp. 163–172 (2004)
9. Sandhu, R., Ferraiolo, D., Kuhn, R.: The nist model for role-based access control: Towards a unified standard. In: Proceedings of the Fifth ACM Workshop on Role-based Access Control, pp. 47–63 (2000)
10. Smith, C.: A survey to determine federal agency needs for a role-based access control security product. In: International Symposium on Software Engineering Standards, p. 222 (1997)
11. Wolter, C., Schaad, A., Meinel, C.: Task-based entailment constraints for basic workflow patterns. In: SACMAT 2008: Proceedings of the 13th ACM Symposium on Access Control Models and Technologies, New York, NY, USA, pp. 51–60 (2008)
12. Ferraiolo, D., Kuhn, R., Chandramuli, R.: Role-Based Access control, 2nd edn. Artech House, Norwood (2007)
13. Herwig, M., Verelst, J.: Normalized Systems: Re-creating Information Technology based on Laws for Software Evolvability, Koppa (2009)
14. Department of Homeland Security Strategic Plan Fiscal Years 2008–2013, Homeland Security, USA (2008), `http://www.dhs.org`

15. Zhixiong, Z.: Scalable role organization based access control and its administration, PhD Thesis (2008)
16. Slone, S.: The Open Group Identity Management Work Area, Identity Management (March 2004)
17. IEEE830:1998, IEEE recommended practice for software requirements specifications. Technical report, Software Engineering Standards Committee of the IEEE Computer Society (1998)
18. Pressman, R.: Software Engineering, A practitioner's Approach, 3rd edn. Mc Graw Hill Book Company, Europe (1992)
19. Kaufman, L.: Data Security in the World of Cloud Computing. Security & Privacy 7(4), 61–64 (2009)
20. Kandukuri, B., Paturi, V., Rakshit, A.: Cloud Security Issues. In: IEEE International Conference on Services Computing, SCC 2009, September 21-25, pp. 517–520 (2009)
21. Shaw, M., Garlan, D.: Formulations and Formalisms in Software Architecture. In: van Leeuwen, J. (ed.) Computer Science Today. LNCS, vol. 1000, Springer, Heidelberg (1995)