An Approach for Flexible RBAC Workflow System

Yuqing SUN, Xiangxu MENG, Shijun LIU, Peng PAN
School of Computer science and technology, Shandong University, China
{sun_yuqing,mxx,lsj,ppan}@sdu.edu.cn

Abstract

With the fast increase of electronic commerce, more and more enterprises and organizations are facilitating their business processes by workflow. To protect information secure and meet the requirement of frequent business changes over time, the security and flexibility become two of the most important aspects that attract attention both from academy and industry. Many related research work on flexible workflow and secure RBAC model were presented respectively. Unfortunately, the analysis and implementation of enforcing RBAC into web-based flexible workflow have not been mentioned. The intention of this paper is to extend RBAC framework further to flexible workflow to support the security, flexibility and expansibility of organization business. A model and its corresponding mechanism are introduced for establishment, dynamical customization and run-time management of the RBAC workflow. A practical system for Property Right Exchange (PRES) based on this model is implemented.

Keywords: flexible workflow, RBAC, access control

1. Background

With the fast increase of web-based electronic commerce, more and more enterprises and organizations are facilitating their business processes by workflow. Workflow management system provides an environment to define, manage and execute business process to improve the performance efficiently. Going with its advantage, it also comes at a cost of increasing information security risk, especially for the unauthorized access to information. According to the statistic from Computer Security Institute (CSI), 71% inspected unauthorized access come from inside and the most serious loss were all caused by internal persons who are authorized and in confidence, not by outside attackers [1,2]。 Hence it is extremely important to establish a security model in web-based system to protect information from unauthorized disclosure and improper action, furthermore to provide supports for ensuring information accessed only for intended purposes [3]. Another increasing problem on workflow is its flexibility to meet the requirements of business frequently changes over time. "Flexibility" means the dynamic workflow adjustment, new workflow addition, the uncertainty characters of workflow such as uncertain workflow branch, uncertain tasks components and uncertain users before a workflow instance in confirmed etc. So a flexible secure workflow for an organization becomes highly important.

The well-known security administration approach has been the Role-Base Access Control (RBAC) model [4] over the past few years, which has been used in a variety of forms for computer system security. Instead of specifying access rights for each individual user, access authorization to objects, called permission or privilege (including object and operation on it) [15], is assigned to roles. Role is an abstract description of behavior and collaborative relation with others in the organization, which is comparatively steady and effective. Roles design depends upon an organization's business activities and permission interpretation, and should reflect the dynamic adjustment of business. A mechanism should be provided to support not only for delivery but also for timely revocation of permissions beyond the required performance period.

The related research work was increasingly developed. The Task-Based Authorization Control (TBAC) model was introduced to provide the notion of just-in-time permissions. It focuses on the processing states and life cycle of authorizations that could enable the granting, tracking and revoking of permissions to be automated and coordinated with the processing of various tasks [5]. A task-role-based control (T-RBAC) model was recently proposed [8,9] in which the permissions are not directly assigned to roles, but assigned to related tasks, and tasks are assigned to related role. Bertino etc. introduce a Temporal-RBAC (TRBAC) model to support periodic role enabling and disenabling expressed by role trigger [10]. He also proposed a model for specifying and enforcing authorization constrains for WFMS [6,7]. Gail-Joon etc. experimented to inject RBAC to secure an existing web-based workflow system [11]. Also there are many research results on flexible workflow recently [13,14,15,16,17]. P.heinl etc presents two approaches to achieving the flexibility in workflows, namely flexibility by selection and flexibility by adaptation [12]. Halliday etc. describe supporting flexibility in distributed workflow [19]. Muehlen gives an overview

of the organizational aspect of workflow technology [18].

Unfortunately, the analysis and implementation of supporting flexible RBAC web-based workflow have not been mentioned. The intention of this paper is to put forward a secure model and corresponding mechanism for flexible workflow. It can dynamically bind users, roles, activities and permissions together. A practical system based on this model is implemented.

This paper is organized as follows. Next section presents the preliminary knowledge of RBAC and workflow our work based on. In section 3, the main result of this paper - flexible RBAC workflow model is presented. Section 4 is the implementation of the practical system for Property Right Exchange (PRES). At last conclusion with direction for further research is made.

2. Preliminary knowledge

2.1. Workflow management

Workflow Management System (WFMS) is used to coordinate and streamline business process in an organization or enterprise. The flexibility of workflow can be implemented at design time and at execution time. The former is achieved by ensuring that there are a number of execution paths through the workflow process so that an appropriate path could be selected for each instance taking account of the circumstances. For the later, the workflow process could be altered by addition, modification or deletion of execution path.

Workflow model is the formalized description of business process that consists of various well-define activities carried out by users according to organization rules. Generally, roles represent organizational agents to perform certain job functions for intended purpose. Users are assigned to appropriate roles based on their qualifications and responsibilities [6]. Workflow instance is used as an instance of workflow model, namely an execution of business process. Every workflow instance is corresponding to a workflow model, while workflow model can derive many workflow instances [13].

2.2. The RBAC96 model

The RBAC96 [4] is a general model for role-based access control (RBAC) which becomes a widely cited authoritative reference and is the basis of current standard developed by National Institute of Standards and Technology (NIST) [20]. The main components of RBAC96 are users, sessions, roles, role hierarchy, permission, user role assignment relationship (URA), permission role assignment relationship (PRA) and constrains. Constraints are used to reflect the policy of organization. RBAC model supports the following well-known security constraint principles:

Least privilege (LP): Only assigning the minimum permissions for the role needing to complete a task.

Separation of duty (SoD): Formulating multi-person control policy to discourage fraud by spreading the responsibility and authority for an action over multi-people. It requires that two or more different people be responsible for the completion of a task or set of related tasks. The approach realized at design time is called Static SOD, while the Dynamic SOD is implemented at run time.

Mutual exclusive (ME): mutually exclusive roles should be assigned to different users. One user is not allowed to have exclusive roles (Static SOD), or one user can have exclusive roles but not allowed to perform both at the same time.

Cardinality restrict (CR): the number of users assigned to a certain role is limited (Cardinality), also the number of permissions assigned to a certain role is limited.

Due to the length of the article, the other description of the RBAC96 is omitted. Details can be got in the reference [4].

3. Flexible RBAC workflow model

The flexible RBAC workflow model is detailed described as follow starting by various terms.

Object: is processed by a user in an activity, can be a database or a sheet, a file, even an item in a form, different according to the idiographic requirements.

Permission: includes object and processing on it highly depending on concrete system.

Activity: is the sub-division, independent, reusable, easy manageable, finely intelligible granular action with logical sequential character performed by definite roles

Workflow: a set of logical, related and sequential activities. Business process is engaged in it.

Sequential relation ($\leqslant$): the performance sequence between two activities. If activity act1 should be performed before act2 then it is denoted in act1$\leqslant$act2.

Some terms and denotations of RBAC are adopted here: U-set of users, R-set of roles, P-set of permissions, A-set of activities, AS-set of activity states, defined as AS={initial, prepared, executing, done, committed, aborted}.

3.1. Activity model

Definition1 (activity) An activity *atv* is defined as a 5-tupls

<p_set, r_set, ant_set , cur_sta, a_type>

where *p_set* is the set of permissions which could be performed in the activity, *r_set* is the set of roles who are permitted to perform this activity that satisfy static SOD, *ant_set* is the set of its direct ancestor activities

that satisfy sequential relation in any workflow, *cur_sta* is the current state of activity, *cur_sta*∈AS, *a_type* is the type of activity including sequential and repeated.

Accordingly, the following mapping functions are introduced:

PriATV(*atv*): *atv* → 2^A, gives the set of ancestor activities that should be processed before the given activity in any workflow.

Roles(*atv*): *atv* → 2^R, gives the set of roles who can perform the given activity.

Activity restricts PRA to be valid only during the activity's lifetime, namely permissions are enabled in the period of activity execution and revoked by the end. The security constraints of LP and static SoD are adopted when designing activities. The reasoning and proof of its logicality property will be given later together with the workflow model. After all activities are defined, the set of permissions assigned to a *role* could be counted by following function:

Perms(*role*): *role*→2^P, gives the set of permissions that the given role have been assigned to.

3.2. Workflow model design and assembly

Definition 2 (workflow model) A workflow model *wfm* is a set of activities defined as a 3-tupls

<atv_set, ≤, ρ>

where (*atv_set, ≤*) is a finite partial ordered set of activities, with a least element *bgn* and a greatest element *end* assuming that *bgn≠end*. *ρ*: A→N is a function used for a repeated type activity that indicates how many times the activity should be executed in workflow.

The following mapping functions are given:

AtvSet(*wfm*): *wfm*→ 2^A, gives the set of activities to form the given workflow model

SeqRel(*wfm*): is a Boolean function to testify whether all the activities in the workflow model are sequential (topological). It is true only for any activity *act1*, act2□AtvSet(*wfm*), if *act1* ≤*act2* then *act1* is arranged in front of *act2*.

Property (sequence constraint) if activity *atv1* is ancestor of *atv2* then *atv1* should be performed before *atv2* in any assembled workflow.

To ensure this property, principle1 should be adopted when assembling a workflow.

Principle1 (permissive activity) An activity can be assembled in a workflow only when all its ancestor activities have been assembled in the workflow before.

A workflow model can be dynamically assembled with sequential activities at run time so as to satisfy the real time customization purpose. The static SOD and mutual exclusive principles are implemented by the functions *AtvSet*(*wfm*) and *Roles*(*atv*) at the same time. Its logicality property is ensured with *SeqRel* and *PriATV* functions according to principle1. It can be explained with a topological graph as follows:

Each activity *atv* in set A is regarded as a vertex of graph. The sequential relation is treated as a directed edge from its ancestor activity *fa* pointing to *atv*. So the directed graph can be treated as an AOV network. If all the vertexes could be topologically sorted, it is proved that there is no circle in the graph, namely all activities in the workflow model can be processed with normally end. The detection is implemented both in static period of activity design and dynamic period of assembling the workflow with activities. An example is given showing how it works.

Let A={a1,a2,a3,a4,a5,a6} is the whole set of activities. If the direct ancestor activities of each are defined as: PriATV(a1)= {}; PriATV(a2) = {a1,a4}; PriATV(a3) = {a2}. Suppose when defining a4, the ancestor activities of it can't include a3, namely a3 ∉ PriATV(a4). Otherwise there would be a circle existing among a2, a3 and a4. So let PriATV(a4) = {}; PriATV (a5) = {a2}; PriATV (a6) = {}. The topological graph of A is shown as figure 1a. According to the principle1, the customized workflow models in figure 1b and 1c are rational while workflow3 and workflow4 are illegal because a5 requires a1, a2 and a4 be processed before it and there is a logical circle in figure 1e. This detection is carrying out in real time.

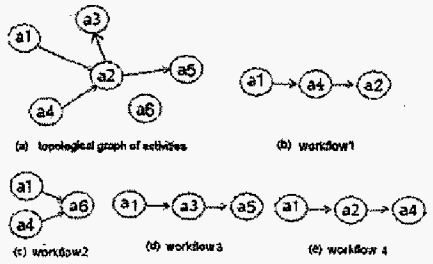

Figure 1 Logical graph of activity and workflow

3.3. Workflow instantiation

Definition 3 (workflow instance) A workflow instance ins is defined as a set of 5-tupls

<wfl_mdl, atv, ≤, r, u>

where *wfl_mdl* refer to the workflow model from which the instance derived, (*atv, ≤, r, u*) is a finite partial ordered set of activities with the definite user corresponding to the permitted roles.

The following mapping functions are introduced accordingly:

RSet(*ins*): *ins*→ 2^R, gives the set of permitted roles in the given workflow instance

USet(*ins*): *ins*→ 2^U, gives the set of users in the given workflow instance

Wfl_Ins(*wfl_mdl*): gives all the executing instances derived from workflow model *wfl_mdl*.

Consistency(*ins*): is a Boolean function to testify whether all the roles in *RSet(ins)* and all users in

USet(ins) for the given workflow instance satisfy the principles of dynamic SoD, ME and CR.

Principle2 (complete exclusion) If two roles are defined as mutually exclusive and a user is authorized to one of them, the user must not access any permission that has been assigned to another.

Workflow instance is processed at run time to actualize the dynamically binding users to roles and permissions. The security constrains are implemented according to principle2 by function *Consistency(ins)* and *Perms(role)*. The determinate user who performs each activity is recorded when workflow instance is generated and executed.

The integrated architecture of flexible RBAC workflow model is shown as figure2. There are four levels. From bottom to top are permissions, workflows, roles, and users denoted with level 1-4. Two definite differences with other models are: permission in this model is not directly assigned to role, but capsulated in activity definition corresponding to current business, and activity has both separated and correlative characters so that workflow can be dynamically assembled with them to satisfy the flexibility. A mechanism to support the model is provided as follow:

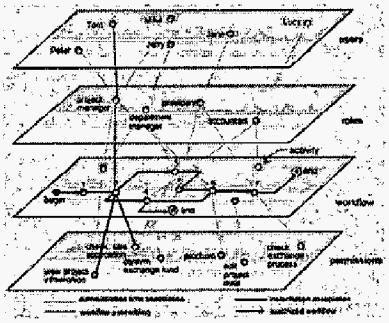

Figure 2 Architecture of flexible RBAC workflow model

Step1 To define activities including permissions which is shown as points in level 2 and thin line between level 1 and 2. Characters of sequence and coherence are considered while defining.

Step2 PRA is implemented by dint of activity to bind roles and permission showing as thin line from level1-3. It should depend upon an organization's actual business.

Step3 URA is defined showing as thin line from level3-4.

Step4 Workflow model could be assembled with activities in definition period or in execution time according to the above principles and shown as the arrow line in level3 and the bold zigzag line between level 1 and 2.

Step5 workflow instantiation is actualized in execution time for real time requirements shown as the medium line.

The flexibility and security constraints are implemented in the process of workflow model customization and workflow instantiation.

4. Implementation

In this chapter an example based on the above flexible RBAC workflow model is presented which is on the background of web-based Property Right Exchange Systems (PRES) for Shandong Province Property Right Exchange Center (SDPREC).

4.1. PRES's requirements analysis

The business of SDPREC covers a large area that include property rights exchange of state owned enterprise and collective-owned enterprise, shared rights exchange of limited company, the intangible assets (intellectual property rights etc.) exchange, surrogate of reforming and recombining enterprises, witness of the property rights exchange process and many other businesses with Chinese features. The workflows of them are quite different and complicated which may be changed at any time based on the state policies and businesses development of SDPREC. The security and expandability are extremely important for them.

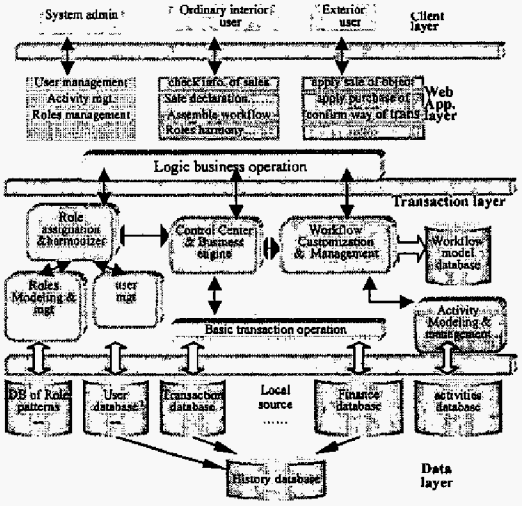

Figure 3 Architecture of PRES

PRES is designed as an integration of web-based property rights exchange system and interior OA system. It involves a large number of users who want to share and grant access to a huge number of security objects. The users include both exterior registered web users and interior authorized employees. Every interior user takes on a few duties whose function and ability

would be dynamically assigned and adjusted according to the organization business.

Aiming to the overall objective of developing a general, flexible and extensible environment for administration and application, the conceptual architecture is proposed as figure 3. Data layer stores the permissions composed of objects and processing. Objects in this system include database, sheet of a database, file, items in a form etc. Transaction layer is the core layer encapsulated the important modules: user management, role modeling and management, activity modeling and management, workflow customization and management, control center and business engine. Detailed description of this layer will be given follow. Web application layer is the operation interface of user.

4.2. User and role management

User management modular is responsible for establishing and managing user database that include organizations of SDPREC and personal information. The PRA is designed in role modeling and management modular (Figure 4). Least privilege, static SOD, mutual exclusive, cardinality restricts etc. security principles are adopted here. But the assignments are not settled, they would be restricted in activity design (in 4.3) to ensure the activation of allocated permissions be reasonable in current business.

Figure 4 Screen snapshot of PRA process in PRES

4.3. Activity design

Activity in PRES is extracted from all business processes with constrains. It likes a restricted template to enable the PRA and URA active only in proper period so as to assure user do right thing at right time. Activity design is implemented in module of activity modeling and management that include defining ancestor activities, selecting licit roles in this activity etc. The roles and permissions can be chosen here only for the authorized those in 4.2. When a role is chosen, its responding authorized permissions are shown below for chosen. See figure 5.

4.4. Workflow management

Workflow customization and management module provides a visual environment for assembling the activities organically based on trade model in execution time and managing the workflow database. The control center and business engine module provides a run-time environment to create and execute a workflow instance. It implements the dynamic user binding of the authorized roles and users, namely Real-time URA.

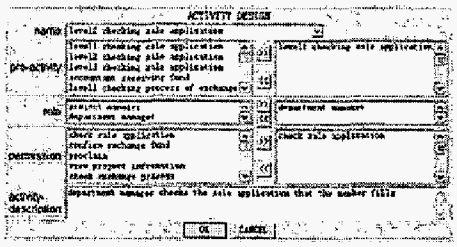

Figure 5 Screen snapshot of activity design

5. Conclusion

In this paper a flexible web-based RBAC workflow model is presented to support security, flexibility and expansibility of organization business. A new mechanism is developed for dynamically customizing the workflow and binding users, roles, activities and permissions. Based on this model, a practical system for Property Right Exchange is implemented which provides an opening, manageable, reusable platform according to the SDPREC business and an integrated environment for performance and maintenance. This model can be deployed into several application domains especially for collaboration environment and complex electronic commerce system. In future work, the model can be extended to hierarchy roles architecture.

Acknowledgment

This work was supported by the National Nature Science Foundation of China (60373026), the National High Technology Research and Development Program (863 Program) of China (2003AA414310) and Science Development Plan Program of Shandong province (2004GG2201131). Hui LIU, Fang YIN and Liang FENG played key roles in the development of the practical system.

References

[1] K.M.Jackson, J.Hruska, D.B.Paker, *Computer Security Reference Book*. Butter-heinemann Ltd., 1992.

[2] Power, T. Tangled Web, *Tales of Digital Crime from the Shadows of Cyberspace*. Que Macmillan Publishing, Aug.31 2000

[3]R.holbein, S.Teufel, "A Context Authentication Service for Role-based Accedd Control in Distributed System – CARDS". In *Proceeding of IFIP 11th International Conference on Information Security*. 1995

[4] Ravi S.Sandhu, Edward J.Coyne, Hal L.Feinstein, and Charles E.Youman. "Rose-based access control model [J]". *IEEE Computer*, Feb.1996,29(2):38~47

[5] Thomas,R.K. and Sandhu R., "Task-based suthorization controls(TBAC) " *proceedings of Computer Fondations Workshop* X,1997

[6] Bertino,E., Ferrari,E. and Atluri,V. "A Flexible Model For The Specification And Enforcement Of Authorization Constrains In Workflow Management System". *Procedings of the Second ACM Workshop on Role-Based Access Control*,1997

[7] Bertino E.,Ferrari E.,AND Atlur V., "An Approach for the Specification and Enforcement of Authorization Constrains in Workflow Management System " ,*ACM Transactions on Information System Security*, Februry 1999,Vol.1,No.1.

[8] S.Oh and S.Park, "Task-role based access control (T-RBAC): An improved access control method for enterprise environment," *Lecture Note in Computer Science 1873,Database and Expert Systems Applications, Proceedings of 11th International Conference*, DEXA 2000,2000,pp.264-273

[9] S.Oh and S.Park, "An integration model of role based access control and activity-based access control using task," *Proceedings 14th Annual IFIP WG 11.3 Working Conference on Database Security*, Aug.2000,pp557-569

[10] E. Bertino and P. Andrea Bonatti, " TRBAC: A Temporal Role-Based Access Control Model", *ACM Transaction on Information and System Security*,Vol.4,No.3,Aug. 2001,pp191-223

[11] G.-J. Ahn, R. Sandhu, M. Kang, and J. Park. Injecting RBAC to Secure a Web-based Workflow System. *Proceedings of 5th ACM Workshop on Role Based Access Control*. ACM, Berlin, Germany, July, 2000.

[12] P.heinl, S.Horn, Jablonski, J,Neeb, K.Stein, and M.Teschke, "A Comprehensive Approach to Flexibility in Workflow Management Systems", *Proc. Joint Intl. Conf. on Work Activity Coordination and Collaboration, WACC'99*, San Francisco, Feb, 1999, ACM Software Eng, Notes, March 1999

[13] Li Hong-Chen,Shi Mei-Lin, "Workflow Models and Their Formal Descriptions", *Chinese Journal of Computer*, Vol.26 No.11, Nov.2003,1456~1463

[14] Zhao W,Hu WH,Zhang SK,Wang LF. "Study and Application of a workflow meta-model". *Journal of Software*,2003,14(6):1052~1059

[15] WFMC. *"Workflow Management Coalition: Terminology & Glossary"*. WFMC-TC-1011, 1999.2. Http://www.wfmc.org]

[16] P.J. Mangan and S.Sadiq, A constraints specification approach to building flexible workflows. *Journal of Research and Practice in Information Technology*, 2002

[17] J.Wainer, etc. "Tucupi: a flexible workflow system based on overridable constraints", *Proceedings of the 2004 ACM symposium on Applied computing*, 2004

[18] Michael Zur Muehhlen, "Organization Management in Workflow Appplication" *Information Technology and Management Journal*, pp271-291, 5(2004)

[19] J.J. Halliday, S.K. Shrivastava and S.M.Wheater, "Flexible Workflow Management in the OPENflow system", *Proceedings of the 5th IEEE/OMG International Enterprise Distributed Object Computing Conference (EDOC 2001)*, Seattle, Sep. 2001, pp. 82-92

[20] Ravi S.Sandhu & David Ferraiolo. "The NIST Model for role-based access control: towards a unified standard[S] ", *5th ACM workshop on RBAC*,2000:47~63