Aquaculture Access Control Model in Intelligent Monitoring and Management System Based on Group/Role

Qiyu Zhang, Yingyi Chen, Zhumi Zhen, Jing Xu, Ling Zhu, Liangliang Gao, Yanzhong Liu

Aquaculture Access Control Model in Intelligent Monitoring and Management System Based on Group/Role

Qiyu Zhang [1,2], Yingyi Chen [1,5], Zhumi Zhen [1,5], Jing Xu [1,5], Ling Zhu [4], Liangliang Gao [1,6], Yanzhong Liu [3,*]

(1.College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, P.R.China

2. Yantai Academy, China Agriculture University, Yantai 264670, P.R.China;

3. Institute of Information for Agriculture, Shandong Academy of Agricultural Sciences, Jinan, 250100, P.R. China

4. Shandong Institute of Business and Technology, Yantai 264605, P.R.China

5. Key Laboratory of Agricultural Information Acquisition Technology, Ministry of Agriculture, Beijing 100083, P.R. China

6. College of Information Science and Engineering, Shandong Agricultural University, Taian 271018, China)

Abstract: Aquaculture intelligent monitoring and management system (AIMAMS) is a web information system covering businesses, home users, and management of aquaculture information, which have many users. The information security and safety equipment of different users is an important issue that we have to consider. Role-based access control introduces the roles concept into user and access rights, and its basic feature are that divide roles depending on the security policy, assign operating license for each role, and assign roles to each user. The user can access the specified object based on their respective roles. In order to achieve different user needs for security, aquaculture access control model based on group/role is developed which is based on analysis of the role-based access control model. Aquaculture intelligent monitoring and management system consists of system administrators, business users, family farming users and technicians, each category contains a number of other users who have different permissions. For such a complex distribution of competences, the user groups are introduced here on the basis of idea of role-based access control model. Users are divided into four groups which correspond to four categories of users, each group is given the largest collection of operational authority, and the users of each group are assigned different roles, which have all or part of privileges of the current group. Technicians can access business and family farming user's information, but can not view some sensitive data, such as price, therefore, technicians group is controlled by field-level permissions of data tables, the sensitive fields of business and family farming user data tables is shielded. This model can effectively reduce the overhead of rights management, and can improve system's scalability. Along with the needs of business development, the system can easily add new user groups and assign permissions and roles for them. This aquaculture access model is universal, it is not only suitable for aquaculture intelligent monitoring and management system but also has reference value for other systems.

Corresponding author: Liu Yanzhong, Email: lyz1228@163.com

Keywords: group; role; access control; aquaculture; permission

Aquaculture intelligent monitoring and management system(AIMAMS) real-time online monitoring the important environmental factors of water quality such as water temperature, dissolved oxygen, PH value in the aquaculture process, by the integration of water quality sensors, Wi-Fi, GPRS and other technologies. This system provides monitoring and management functions which is data query, data collection, curve analysis, and provides more functions which is aquaculture profiles, map browsing, disease prevention and control, feed fed decision, application configuration, provides comprehensive information intelligent automation services for aquaculture users。AIMAMS is a java-based web information system which covers business and many home users. How to ensure information security and the safety equipment of different users? Role-based access control (RBAC) is a good choice. As a security mechanism，RBAC has been widely accepted, which can greatly reduce the cost and complexity of large networked and Web-based systems[1].

1 Role-based Access Control

The concept of RBAC began in the 1970 s，accompanied by multi user and multi application online systems[2]. The basic idea of RBAC is that introduce the concept of role between the user and the permission and link users and roles, and then control user to access system resources by giving permission to role. Role is a set of permission, a user has a role means the user can have all permission owned by the role. A user can have many roles and a role can be assigned to many users; A role can contain multiple permission and a permission can be included more than one role. A user is not directly associated with permission but get permission by role.[3]

To facilitate understanding of the various aspects of RBAC, Sandhu et al[2] defines four models, their relationship are shown in Figure 1, Figure 2 describes the basic features.

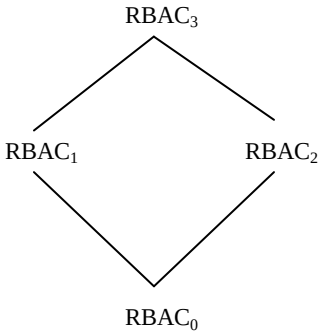

Figure1 Relationship among RBAC models

1.1 $RBAC_0$[2]

$RBAC_0$ is the base model of RBAC, which contains the most basic requiremets. $RBAC_0$ model is composed of various parts of Figure 2.

$RBAC_0$ has the following components:

- U, R, P, and S(users, roles, permissions and sessions),
- $PA \subseteq P \times R$, many-to-many assignment relation to permission and role,
- $UA \subseteq U \times R$, many-to-many assignment relation to user and role,
- user: $S \rightarrow U$, function mapping user (s_i) of each session s_i to the single user (user

(s_i) is a constant in the session's lifetime)

- roles: $S \to 2^R$, function mapping roles(s_i) of each session s_i to a set of roles, roles(s_i) $\subseteq \{r|(user(s_i), r) \in UA\}$ (which can change with time) and session s_i has the permissions $\cup_{r \in roles(s_i)} \{p|(p, r) \in PA\}$

In this model, a user is an organization's staff. A role is a function of a task or job title within the organization. An permission is an approval that users access and operate data and resources in the system. As a mapping from one user to many roles, session can be created during the user activates some subset of their roles.

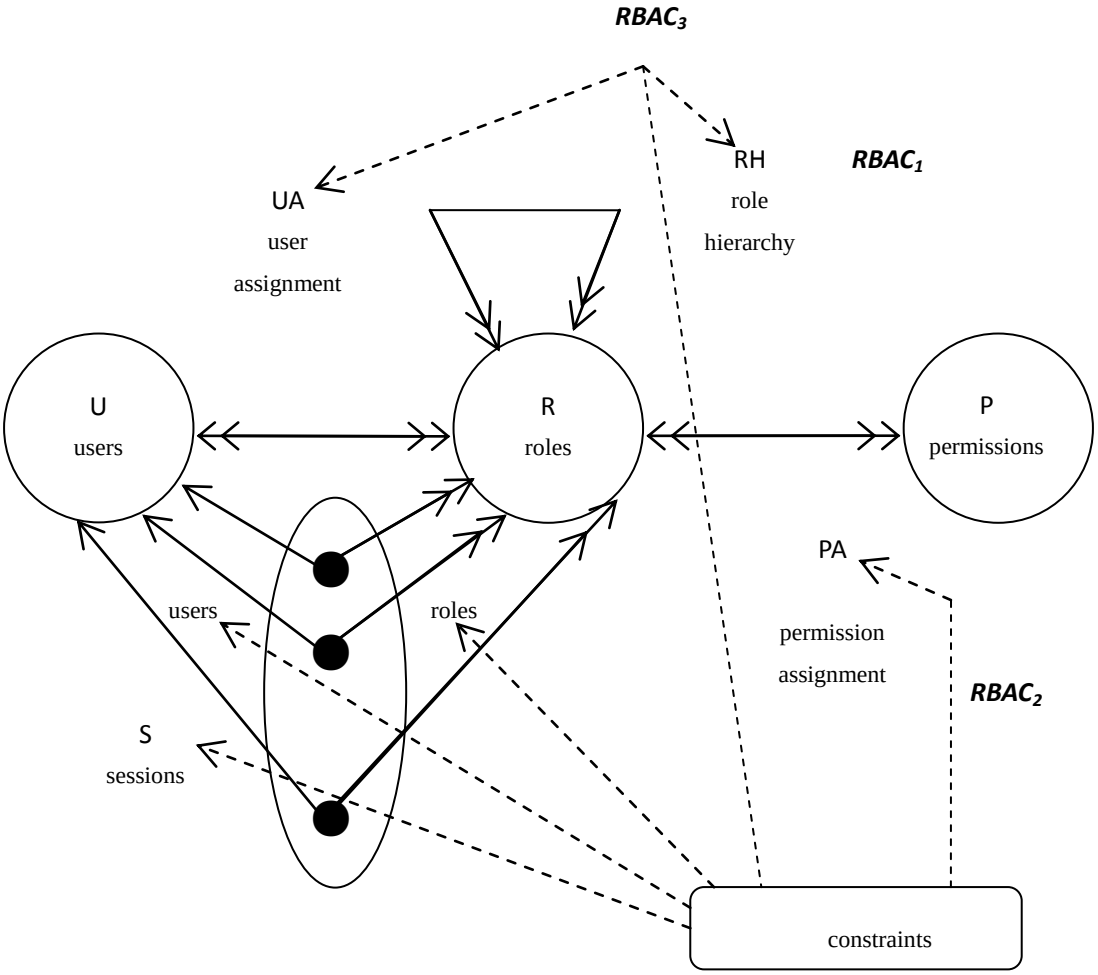

Figure 2 RBAC models

1.2 RBAC$_1$[2]

Based on RBAC$_0$, RBAC$_1$ increases the concept of role hierarchy that roles can inherit permissions from other roles.

The RBAC$_1$ model has the following components:

- U, R, P, S, PA, UA, and user are unchanged from RBAC$_0$;
- $RH \subseteq R \times R$ is a partial order on R called the role hierarchy or role dominance relation, also written as $\geq$;

- roles: $S \to 2^R$ is modified from $RBAC_0$ to require roles(s_i)

  $\subseteq \{r|(\exists r' \geq r)[(user(s_i), r') \in UA]\}$ (which can change with time) and session s_i has the permissions $\cup_{r \in roles(s_i)} \{p|(\exists r'' \leq r)[(p, r'') \in PA]\}$

1.3 $RBAC_2$[2]

Based on $RBAC_0$, $RBAC_2$ increases constraints mechanism(that is, configuration is limited between different components in RBAC). These constraints are needed to determine the values of various parts in $RBAC_0$ are acceptable or not. Only those values which are acceptable is permitted.

1.4 $RBAC_3$[2]

$RBAC_3$ combined with $RBAC_1$ and $RBAC_2$, provides both role classifications and hierarchies.

2 Aquaculture Access Control Model Based on Group/Role

The paper studies the role-based access control model, and combines with the characteristics of aquaculture intelligent monitoring and management system, then proposes the aquaculture access control model based on group/role. The model divides into four major groups based on the nature of users, and each large group divides into several different groups by per unit or duties. Each group has a largest collection of identified operational authority. A manager is assigned to a group, who assigns different roles to each user group, the user has all or part of the privileges of the current group.

2.1 User Groups

AIMAMS has many users. Depending on the subordinate units, the users divide into four categories which are system administrators, business users, home users and technicians. System administrators are staff who work in system website operator unit, business users are enterprise who use the system, home users are individual users who use the system, and technicians are users who provide technical guidance to business users and home users, for example, technicians of fishery technical extension station and so on. Except system administrators, each type of user has users coming from different units, these users also contain a number of users who have different functional privileges. For such a complex distribution of competences, the paper introduces the user groups basing on the basic idea of role-based access control model, and divides the four categories into four groups. Each users group divides into different groups by unit.

The group of users G={GA,GE,GF,GT}， where GA={GA_0, GA_1, GA_2,…, GA_m} is system administers group. GA_0 is super administrator group whose function is to manage other system administrators group; GA_1~GA_m is system administrator group whose function is to manage business user, home users and technicians by region, area can be north, northeast, east and other large areas, can also be provincial, municipal and autonomous regions; each region’s the administrator user is responsible for the maintenance work which contain managing business users group, home users group and technicians group and initializing the system. GE={GE_1, GE_2,…, GE_n} is

business users group, GE_1~GE_n is specific business user groups. GF={GF_1, GF_2,…, GF_s} is home users group, GF_1~GF_s is specific home user groups. GT={GT_1, GT_2,…, GT_t} is technicians group, GT_1~GT_t is specific technician groups who guide business users and home users.

2.2 Users

Because a lot of units involved, the system administrator is impossible to establish an account for each user. A better way is grouped by team management, that is, administrators can specify an administrator user for each group of business users, home users and technicians, and the group administrator can add new user in oriented group.

The user U applies the same packet format as users group G, that is , U={UA,UE,UF,UT}， where UA={UA_0, UA_1, UA_2,…, UA_m}， UE={UE_1, UE_2,…, UE_n} ，UF={UF_1, UF_2,…, UF_s}，UT={UT_1, UT_2,…, UT_t}。As many user groups, one certain business user group UE_i ($1 \leqslant i \leqslant n$) as an example to illustrate users representation. UE_i={ue_{i0},ue_{i1},ue_{i2},…,ue_{ik} }， k is the number of non-administrator users of this group, ue_{i0} is the administrator user of this group.

2.3 Roles

Defining roles is associated with specific applications, these is no fixed pattern, the model defines roles using the “two-step method”[4]. The first step is to determine the class of role, according to the business processes of the business, the commonality of the various stages of the business is abstracted as function terms, in order to define each class of role. The second step is to determine the subclass of role, the model makes sure that the class of user group has permissions (like query, add, update, update, startup, shutdown) to specific data or equipment, according to type of business, users powers, as well as the current state of the system, that is the true sense of role.

They are predefined roles of system, which are determined using above method. For each group, the system pre-defined roles are not directly used, the role of each group must is associated with the data and equipment of current group, and must create a new role by inheriting existing roles or according to the special needs of current group.

The role R applies the same packet format as users group G, that is, R={RA, RE, RF, RT} , where RE={RE_1, RE_2,…, RE_n} , RT={RT_1, RT_2,…, RT_t}. As many user groups, one certain business user group RE_i ($1 \leqslant i \leqslant n$)as an example to illustrate roles representation. RE_i={ re_{i0}, re_{i1}, re_{i2},…, re_{ih} }, h is the number of non-administrator roles of this group, re_{i0} is the administrator role of this group.

2.4 Permissions and Representation

2.4.1 Permissions

Depending on the circumstances of AIMAMS, the permissions are divided into six kinds, they are menu permissions, page access permissions, button permissions, access permissions to data area, field-level access permissions and device permissions.

(1) Menu Permission

In order to make different levels of minimum user access to their own user

interface, the menu is managed as an authority[5]. The system grants the menu permissions, and organizes flexibly application's menu composition based on the user's permissions, then each role user has a better human-computer interface[6]. When the lack of user permissions, the menu item is grayed out[7] or not displayed. The latter is better.

There are two menus in AIMAMS, they are main menu and tree menu for on-line monitoring. The main menu consists of a menu and submenu, and grants permission for the role by clicking the check box of two-dimensional table composed by a menu and submenu, online monitoring tree menu grants permission for the role by clicking the check box of the tree menu. And these two menus dynamically generated according permission.

(2) Page Access Permission

In order to prevent unauthorized access to the page, the system must set permission for each page, so the page verifies access permission when it is loaded initially to allow or ban the user from accessing the page[6]. The system is modular in design, and creates a file for each module, where stores the relevant page files. IO operations traverse through all folders in the root directory of the system, and get each module JSP pages. The tree structure with check box grants permission for the role, which is composed by module and page.

(3) Button Permission

Button operation is an important operating mode bearer service processing actions. By controlling button permission, applications can achieve transaction processing for multiple users at different stages of the same transaction on the same page, then make enterprise information systems more intuitive and efficient[6]. In fact, the button permissions correspond with the SELECT, INSERT, DELETE, UPDATE operation of the data table. When a user has some permissions, the corresponding button is clicked, otherwise it is gray, that can not be clicked.

(4) Access Permission to Data Area

Each group operating pages of business users and home users are the same, but they can only operate their own data and equipment. According to the business process, the page data are divided by group, so that different user can only handle the data that the user has the permissions on the same page[6].

(5) Field-Level Access Permission

When the different users use the same type of permission, there is a difference that they can access resources and use functions[8]. In this case, the system uses the field-level access control method in data table. Technicians can access information of business users and home users, but can not view sensitive information, such as price and value. During view, each field must determine whether the current role be prohibited to display , only field that is not prohibited can be displayed.

(6) Device Permission

Online monitoring system can control aerators, feeding machines and other equipment according to monitor status of the system. In order to ensure the safety of equipment, only the current group user who have permission can startup and shutdown operating position of the devices.

2.4.2 Representation of Permission

The authorization status of role can be described by access control matrix[9]. In the access control matrix, the row is a role, and each column is an authorized ACL(Access Control List), which is a Boolean value, as shown in Table 1. The matrix value of the i-th row j-th column is represented by RP_{ij} , RP_{ij}=TRUE means that the role of R_i has permission P_j, otherwise it does not have the permission.

Table 1 Access Control Matrix

	P_1	P_2	…	P_j	…
R_1	TRUE	TRUE	…	FALSE	…
R_2	TRUE	FALSE	…	TRUE	…
…	…	…	…	…	…
R_i	FALSE	FALSE	…	TRUE	…
…	…	…	…	…	…

Role permission can be saved in HashMap, and only the permission of the access control matrix whose value is true can be saved. There are six permissions, the menu permission separates main menu and online monitoring tree menu, so there are seven access matrix. Button permission, access permission to data area and field-level access permission are combined into a HashMap, which all operate the data table. Therefore, the system can devise five HashMap to save all permissions of a role.

2.5 Constraints

In a model, the constraints are divided into relation constraint, precondition constraint, cardinality constraint, time constraint, address constraint and so on.

(1) Relation Constraint

Conflicting role can not be granted to the same user, for example, in order to prevent fraud, the purchaser role and accounting role can not be granted to the same user; conflicting role can not be activated in the same session[10], for example, the administrator role and other role can be granted to a user, but can not be activated at the same time.

(2) Precondition Constraint

A user wants to be granted the role A, it must already have the role B, that is a role wants have a permission, the precondition is that it must have another permission[10]. If a user already has a technician role, it can be granted a technical director role. If a user already has teh corresponding menu permissions, it can be granted the submenu permissions.

(3) Cardinality Constraint

Cardinality constraint is a upper bound of users number, these users are granted the same role. n represents the maximum number of users assigned this role[11]. The value of n is specified by the group administrator, and the default value is 1.

(4) Time Constraint

Roles or permissions can be only activated in certain time range[12]. When a role need to grant a user temporary, the role can be setted time constraint, and it can be activated only in the specified time range.

(5) Address Constraint

Address constraint is divided into IP-role[13] and MAC-role constraint, that is ,

address constraint binds the IP address, MAC address and the role together. The administrator has a very high permission. The extent of damage to the system caused by their operation is far higher than the average user. If the administrator permissions is to be attacked, the damage to the information system is inevitably fatal[14]. So the administrator can be granted the IP-role constraint, MAC-role constraint or IP-MAC-role constraint, and only users whose IP, MAC address is required can activate the administrator roles.

3 Access Control Process

(1) User login, verify the user name and password. If the user passes the verification, the user is prohibited access system. Once authenticated, users get all the role information, and check relationship constraints, time constraints and address constraints, then get the effective role of the user. If a time constraint of a role in time constraint after the current time, this role is an effective role, but there are time constraints.

(2) If the user has multiple valid roles, the six kinds of permissions for each user role represents a seven access control matrix, access control matrix between the different roles is a logical or operation, the user can get the access control matrix. If the user has a valid role, the user's access control matrix if a role's access control matrix. If the user does not have a valid role, the system gives tips and prohibits the user access.

(3) The access control hash table, which is structured according to the user's access control matrix, is added to the user class, and the user's complete object is added to the session. At the same time, the user name is added to the online user hash table, where the scope is application, the key means user name, value is the flag vale of corresponding seven access control matrix. value=zero means the corresponding access control matrix does not change, value=1 means change. Each time a user generates access control hash table, the value of the corresponding online user hash table is set to "0000000". If the user is logged, the administrator will determine which access changes when the user roles or permissions changes. The value is set to 1, if the matrix is changed.

(4) When the user has access or operation request, the administrator firstly check the online hash table to see whether role or permission is change. If there is no change, turn (5); if there are changes, the system generates access control matrix again, and generates the hash table.

(5) By querying the corresponding access control hash table, the system judge whether the user has permission. Only the user has permission, its access or operation will be response.

4 Conclusion

Based on the research of role-based access control model, the paper put forward the aquaculture access control model based on group/role, combined with the safety requirements of AIMAMS. The model can simplify the complex permission, reduce the system permissions management overhead effectively, and improve the expansibility of the system. With the business development, the system can easily add new user groups and grant permissions and roles to them. The model is general, it is

not only suitable for AIMAMS, but also has reference significance to the other system.

Acknowledgements

This paper was supported by Shandong Province Self-innovation Projects (2012CX90204) and the Shandong ProvinceKey Research & Development Program "Research on aquaculture management and application platform technology based on big data" (No. 2015GGC02066).

[References]

[1]David F. Ferraiolo, D. Richard Kuhn , 2007.Ramaswamy Chandramouli.Role-Based Access Control, Second Edition. Artech House.

[2]RaviS.Sandhu, EdwardJ.Coyne, HalL.Feinstein and CharlesE.Youman. 1996. Role-Based Access Control Models. IEEE Computer,Volume 29: 38-47.

[3]Michael E. Shin and Gail-Joon Ahn. 2000.UML-Based Representation of Role-Based Access Control. IEEE 9th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'00),195-200.

[4]Zhang Wei. 2003.Research on Role-Based Access Control and ITS Application in Court System[D].ChengDu:Southwest Jiaotong University.

[5]Yun Liaofei. 2007.The Application of Role-Based Access Control Technology in Union Equipment Management Center System[D].XiAn:Xidian University.

[6]Xu Jianrui,Chen Dan. 2010.Research of Permission Management Based on Role and Discretionary Access Control[J].Software Guide, 12(9):160-162.

[7]Zhang Haitao Liu Zhifeng Li Yang,et al. 2006.Research and application of Role-Based Access Control in Privilege Management[J].Microcomputer Information, 22(9-3):29-31,96.

[8]Yang Fan,Zhang Bo.RBAC Based Classified Application System Access Control Method Design[A].The Paper Collection of 2012 MIS / S & A Academic Conference[C].2012.182-185.

[9]Tang Peng-xiang,Chen Meng-dong,Liu Lian-zhong,,et al.DESIGN AND IMPLEMENTATION OF A PRACTICAL ROLE-BASED ACCESS CONTROL MODEL[J].Computer Applications,2002,22(12):41-43.

[10]Wang Zhuo,Feng Shan. 2003.Specify RBAC Constraints Using Object Constraint Language[J].Computer Engineering and Applications, 39(21):100-102,109.

[11]Zhang Hong-qi,Zhou Jing,Zhang Bin. 2008.Research of Extension of Static Constraints Mechanism in RBAC Model[J].Journal of Beijing University of Posts and elecommunications, 31(3):123-127.

[12]Dong Guang-yu,Qing Si-han,Liu Ke-long. 2002.Role-Based Authorization Constraint with Time Character[J].Journal of Software ,13(8):1521-1527.

[13]Wu Xian. 2002.Web Applications of Role-Based Access Control [D].DaLian:Dalian University of Technology.

[14]Fan Jin-sheng,Guan Bao-can,Li Xiao-dong. 2008.Design of extended role-based access control model and its implementation.Computer Engineering and Design, 29(18):4178-4721.