Advanced Engineering Forum *Online: 2012-09-26*
ISSN: 2234-991X, Vols. 6-7, pp 273-278
doi:10.4028/www.scientific.net/AEF.6-7.273
© 2012 The Author(s). Published by Trans Tech Publications Ltd, Switzerland.

The Design and Implementation of the Complicated Resource System Based on RBAC Model

Yan Chen [1, a], Chao Luo [2,b], CanYing Huang [1,c], ShangPing He[1,d]

[1] Science and technology college of NanChang University, Nanchang , China

[2] YiJi Network technology company LTD,Fujian , China

[a]290311310@qq.com, [b] vikiluochao@163.com, [c] 674589196@qq.com, [d]309511954@qq.com

Keywords: RBAC model, Resource model, Complicated resource, J2EE

Abstract. As a key access control mode that multiple characters application system is presently used , RBAC(Role-Based Access Control) can solve the problem of dynamic multi-users and multiple characters excellently, but when facing complicated resource types and business processes, it must be expanding on that basis. The article has put forward and realized a permissions model which can solve complicated resource system and business processes—resource model.

Introduction

In the simple small and medium application system, the users and characters is few and have little change, resource types is also quite unity, so when designing permissions model, the traditional base RBAC model can solve the problem well. But when facing the system proposal of enterprise classes, the distributing form of resource is various, and the demand also changes fast, the system is facing difficult problems that how manage complicated resource dynamically and how distribute authority dynamically, and these resources are also associated with one another, form more complex resource types, Now the traditional base RBAC model has failed to keep up with this complicated system.

The article has put forward a permissions model which can solve the said complicated resource system—resource model, and has realized this resource model which can manage complicated resource dynamically and distribute authority dynamically.

Model Design

Basic RBAC Model. The authority design can be described by a simple logical expressions after distilling abstractly, that is to say, "who" have proceeded with how " operation" to "what", if return true, express having operation right limitation, and the reverse is also true. Abstracted model is shown as Fig. 1.

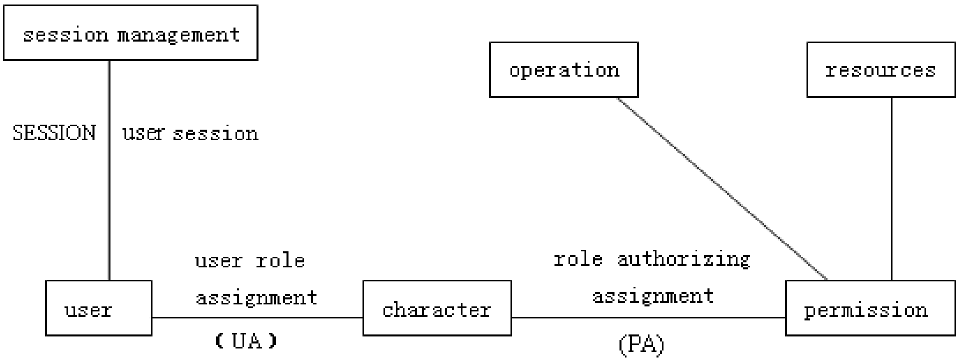


Fig. 1 Basic RBAC model

In Fig. 1, a collection of users expresses characters, that is "who" in logical expressions, resources is "what" in logical expressions, operation and resources have composed of "operation" in logical expressions. session management mainly manages session datas after different users landing, so that whatever web, whatever the embedded, whatever application program on desktop and so on can realize distributed disposition, integrating in one another by authority model. If we have authority distribution massed to center engine to exchange data and verificate manage, it will not only improve efficiency and security, but also realize a real sense of dynamic management.

The Design Of Resource Model. The design and permission of authority model, in the final analysis, is distributing and managing resources, resource is an abstract concept. Very basically put, resources may be a type of data or a terminal equipment, may be also a series of result of calculating, even may be a random combination which is mentioned above, in other words, just you consider it very important ,which is used to distinguish what is different people see or when operating differently, all can be considered as resources. when we design application system, we probably meet even the same kind of resources may arise different problem of authority distribution[1]. In such situations, the resources also fall into the type of irrelevant business and the type of relevant business. The type of irrelevant business can be understood as coarse granularity(express type levels, that is to say ,we only consider the type of object),does not consider the a particular instance of this the type of object, on the contrary, the type of relevant business, is slender granularity(express instance level, that is to say, we need to consider the instance of object). Of course, simplely it can be understood as that the slender granularity is an exception of the coarse granularity. Through the description above, we can separate from each other, although the types of resources is kaleidoscopic, but it can be still abstracted away using the way of data structure[2].

The atomic resources model is shown as Fig. 2. This resources is one of the most common resources types, in actual application, it can be understood as a single functional node, for example: a terminal equipment.

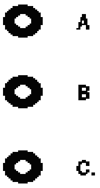


Fig. 2 The atomic resources model

The linear resources model is shown as Fig. 3. The linear resources is the simplest relational resources, each resources are dependent of each other but exist relationship of being brothers, in actual application, areas, categories and so on are all classic linear resources.

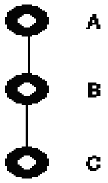


Fig. 3 The linear resources model

The tree resources model is shown as Fig. 4. The tree resources model is a more complex resources type in resources models, there is a paternity among resources, that is to say containment relation, in many application systems, solving this type of problem is all by considering the tree resources as the linear resources to manipulate. In the applications of changed little, this way is not too bad, but meeting the applications that needing high precision, it must proceed with authority selection by the forms of the tree resources. In actual application, the organization structure of department, administrative areas division are all the typical tree resources type.

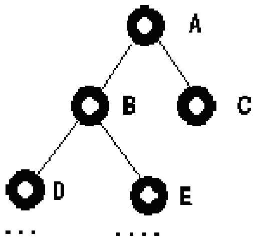


Fig. 4 The tree resources model

The figural resources model is shown as Fig. 5. The figural resources is more complicated resources type, here we mean mainly a directed graph, that is to say, there is complicated uncertain relation between every resources nodes, from starting node to terminal node, it can be have different operations, in actual application, for example, business process is a typical figural resources, because business process may be changed, different starting node, the different resources range which had occurred also differ greatly.

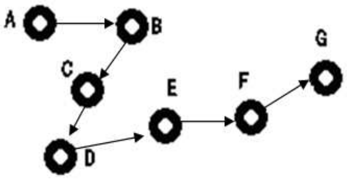


Fig. 5 The figural resources model

The backward resources model is shown as Fig. 6. The backward resources is a backward areas for character which has authority on the basis of the atomic resources, the linear resources and the tree resources. It can be simply described as that all this type of resources can visit except this resources.

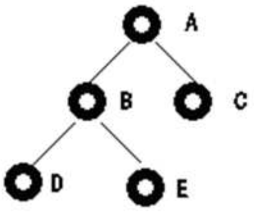


Fig. 6 The backward resources

So far the described resources types are the coarse granularity resources as mentioned above, every node in every type resources all can become slender granularity resources, we can have this resources designed to become slender granularity resources example in order to manage and control for convenience at a relevant business time[3].

The Implementation Of The Model

Take the resources model which mentioned above as the theory model, the implementation way of this model is diversified(such as: the implementation of web way[4]), also can be developed in many developing platform(such as: independent server, the embedded, the distributed), this article will adopt web way to realize, it can expand to other interfaces because it can provide interface modes such as socket, wpf, websevice. Its resizable array is shown as Fig. 7.

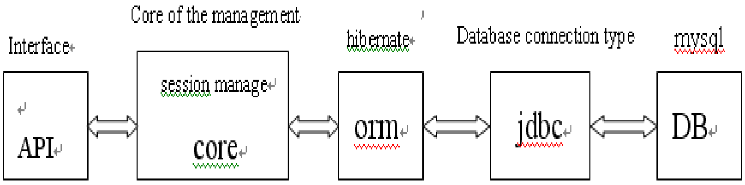


Fig. 7 The block diagram of the model

The implementation of the model can be divided into five layer, it is respectively data layer, business logic layer, object persistence layer, kernel, API layer.

The Data Layer. The major function of the data layer is storing data. The design chooses MYSQL of small relational database management systems which is free, open, supporting many platform , convenient transplant and disposition. The relations diagrams of database design is shown as Fig. 8.

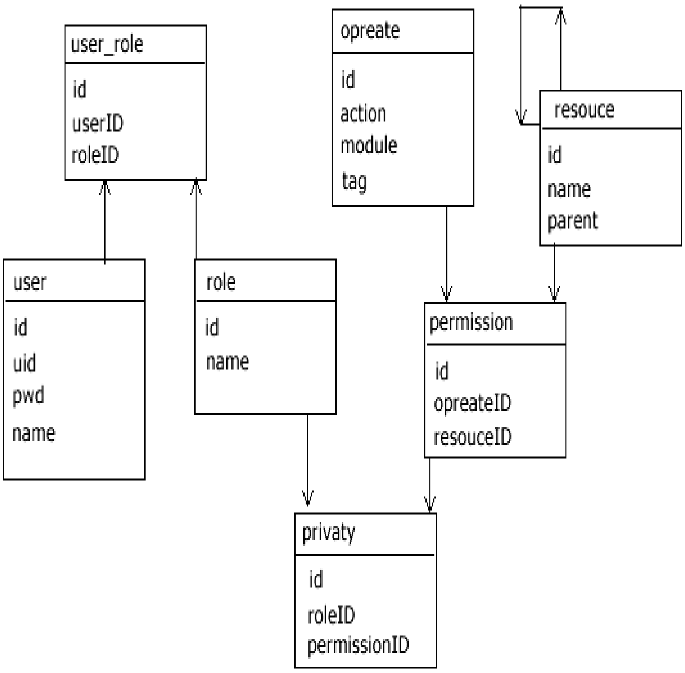


Fig. 8 The relations diagrams of database design

In Fig. 8,resouce is a resources type, it may be the ID number and the corresponding table name which any table saved in permission table, resouce may be any table describing resources type, and supports mapping by itself; User mainly saves user's information table; Role shows character table; Operate shows operating behavior table; Permission is a authority table; Privaty is a authorized table; User_role is a mapping table of user character. Of these tables, user and character is the relationship of one： N; Role and Permission is the relationship of N:N; Operate and Resouce is the relationship of N:N.

The Business Logic Layer. The function of the business logic layer is providing the basic platform for the whole core application. JE22 has powerful object-oriented ability, there are many mature products in this platform as support such as the frame of HIBERNATE, Spring and so on. At present, many systems have been developed on JE22 platform, using the JE22 platform can not only decrease the difficulty of disposition and the cost of intergration, but also improve the compatibility of the system.

The Object Persistence Layer. The object persistence layer(OEM) is used for having relational databases mapping as object-oriented data process framework. The article chooses the hibernate which is object-relations mapping framework of open source, it has powerful persistence processing capabilities and with maturity and stability, and it has also provided a convenient framework for user, from object-oriented domain models to the mapping of traditional relational databases. The article ingeniously converts data storage (relational Data Storage)into object-oriented examples by ORM, has all kinds of operations of complicated abstract data becoming simpler and more convenient.

The kernel. The kernel is the core part of the whole RBAC algorithm, it is in charge of verificafion of authority, it involves in many logical data verification, session management and so on. The session management is mainly used in authorization management of multiple platform, each terminal point can connect the server-side by the way of socket, wpf, websevice and so on. But all of the session management is in the kernel part.

The API. In the achievement process of the API, it contains the two parts of the achievement of resources interface and the achievement of exterior interface.

The resources interface is interior interface, whenever the datas in database how to proceed with storage, it can be turned to resources type example by inheriting this interface, in order to control and use the kernel. Here using an abstract factory pattern to proceed with implementation, the type picture of resources type is shown as Fig. 9[5].

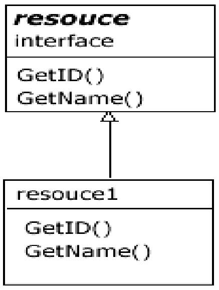
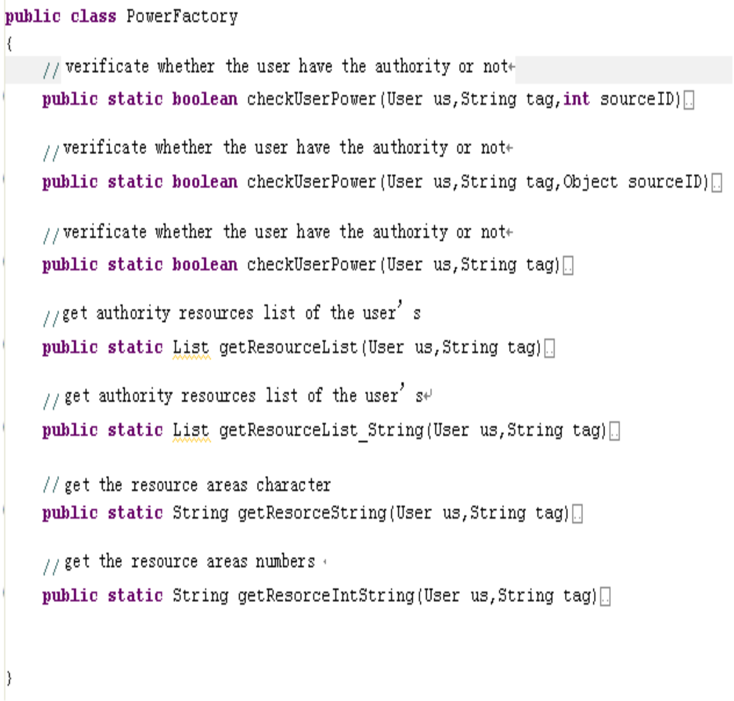


Fig. 9 The type picture of resources type

The exterior interface provides the whole authorized verification and returns the results, all places involved in authorized verification only need to call the interface beneath can return the verification results. The exterior interface is shown as Fig. 10.



```
public class PowerFactory
{
    //verificate whether the user have the authority or not
    public static boolean checkUserPower(User us,String tag,int sourceID)□

    //verificate whether the user have the authority or not
    public static boolean checkUserPower(User us,String tag,Object sourceID)□

    //verificate whether the user have the authority or not
    public static boolean checkUserPower(User us,String tag)□

    //get authority resources list of the user’ s
    public static List getResourceList(User us,String tag)□

    //get authority resources list of the user’ s
    public static List getResourceList_String(User us,String tag)□

    //get the resource areas character
    public static String getResorceString(User us,String tag)□

    //get the resource areas numbers
    public static String getResorceIntString(User us,String tag)□


}
```

Fig. 10 The exterior interface

Whenever in the common system of B/S and C/S, or in the distributed system, it also can call the interface above by the way of socket, websevice, wpf and so on, consequently has realized the authority scheduling and the separation of the authority verificafion node.

Conclusion

The extendable resources model based on RBAC put forward by the article is one of the large number authority models, on propose of solving the problem how to manage and dispatch the complicated resources types of multiply platform and dynamic multiply authority. In the real project(the festival reservation systems of number a know all in Jiangxi)(thirty eight characters, one hundred and fifty people at the same time, fifty five resources nodes, the real-time system of many process businesses), it has solved the management problem of the multiply character, complex resources and dynamic process, and it can be running stably during rush hours, In this way it prove that the model designs is reasonable and successful.

The model designed in the article can be used in network system of distributed disposition, implementing the real centralized management. And if the hardware encryption device is used, it could have be used in superior authority mechanism system (such as bank, industrial control).

References

[1] Alan Shalloway, James R Trott: *Design pattern and analysis(the second edition)* (people post and telecommunications press, China 2006),p. 127-129

[2] Information on http://blog.chinaunix.net/u1/57965/showart_469429.html ,2010

[3] L.F.Fan, G.Y. Lin:*Program designing tutorials of Java Web*(people post and telecommunications press, China 2010), p.177-189

[4] Roger S .Pressman:*The research method of software engineering practitioner* (china machine press, China 2006), p.58-64

[5] W.Si, S.X.Feng:*Data structure and algorithm analysis(the second edition)* (china machine press, China 2004), p.121-145