**以RBAC模型为基础来分析B端权限系统的设计思路（业务技能）**

**前言**

       工作中做过一些权限管理的产品，每次都有总结，但每次收获又有不同，都有新的理解。经过经验的积累，知识的丰富，慢慢的发现主流的权限管理系统都是RBAC模型（Role-Based Access Control 基于角色的访问控制）的变形和运用，只是根据不同的业务和设计方案，呈现不同的显示效果。本文主要从以下几个方面进行整理说明：**RBAC模型的解读与说明、角色权限系统实例分析、设计角色权限系统时的几条建议**。

**一、RBAC模型的解读与说明**

**1.1  角色权限系统的理解**

       说明RBAC模型之前，问我们自己一个问题，为什么我们要做角色权限系统？角色权限系统最终能实现哪些效果？一个很明显的答案就是平台存在不同权限的用户，业务要求不同人应该使用不同的功能、查看不同的内容。这个答案中就包含了，角色权限系统的最重要几个元素：用户----角色----权限。

      是不是所有平台都需要建立角色权限系统呢，是不是所有角色权限系统给规律都一样呢。答案肯定是否定的，但是所有的角色权限系统有据可依只是一定。接下来我们就介绍下，角色权限系统中重要一个基础内容——RBAC模型。

**1.2  RBAC模型是如何承载角色权限系统**

      RBAC全称是基于角色的访问控制。角色是这个模型核心，角色往前可以和用户关联，让用户拥有角色属性；角色往后可以和权限关联，让权限有个归类代表。它们之间的关系如“图1.1用户角色权限关系表”所示



                                                图1.1  用户-角色-权限关系表

       通过上图有人会问为什么不直接给用户分配权限呢，为什么要有角色这一环节。其实图中角色表达的是传递的意思，也就是说可以通过角色来关联用户和权限，同时也可以，直接给用户配权限。只是直接给用户配权限，少了一层关系，扩展性就弱了一些，比较适合那些角色类型少的平台。具体案例后文有具体说明。

       当用户基数很少时，我们可以直接给相应用户指定相应角色；当平台角色类型很少时，我们也可以直接在角色上挂靠用户。但是当平台用户基数增大，角色类型增多时，如果直接给用户配角色，管理员的工作量就会很大。这时候我们可以引入一个概念“用户组”，就是将相同属性的用户归类到一起。具体关系如“图1.2 用户组-用户关系表”所示。

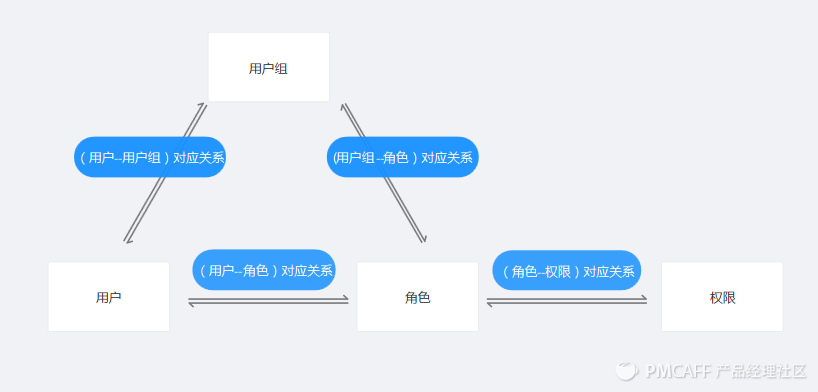


                                                   图1.2 用户组-用户关系表

       如何理解相同用户属性的用户属于一个用户组，这里的用户属性是什么意思。其实这里的相同用户属性是一个泛词，就是同一部门、同一项目组、同一岗位等等这些信息。具体如何归类，就看具体业务需求是如何规定的。具体的归类方法，后面案例给了几个方式。

      有了用户组，我们就可以直接给用户组配置权限，这样整个用户组的用户就拥有了相应的权限。

      图中还有两个关键词“（用户--用户组）对应关系”和”（用户组--角色）对应关系“。在这里就先解释一下。“（用户--用户组）对应关系”就是说，用户按照性别，按照地区，按照部门，按照岗位，按工作内容，或者按照职权等等组合在一起。“（用户组--角色）对应关系”就是给用户组配角色，但是这里要注意，一个用户组可以有多个角色（比如张小明、王小刚、李小红既是用户体验部的成员，同时又是项目A的交互设计师）。

     有了用户，有了角色，那权限我们又如何理解呢。详情见“图1.3 用户-角色-权限关系表”。

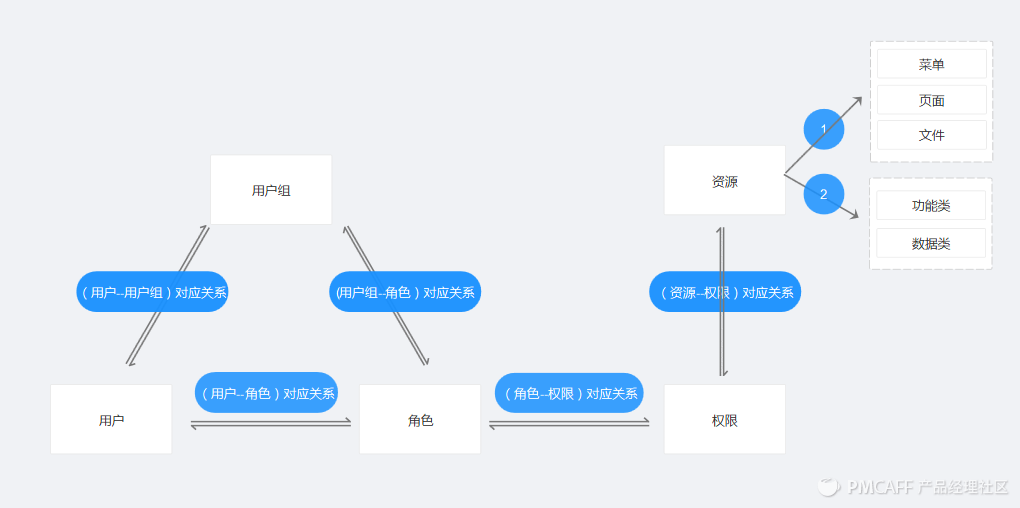


                                                      图1.3 用户-角色-权限关系表

        权限是资源的集合。这里的资源指的是软件中所有的页面、模块、标签、文件、功能、字段等等。具体的权限配置上，目前有多种分类方式，有的人按照功能类、数据类、字段类进行分类整理；有的人就直接把页面、标签这些直接当成权限详情进行配置。不管哪种方式，都是将软件权限根绝业务需求进行分类。

      以上是我根据自己的工作经验对RBAC模型的理解。这是一种比较通用、详细的规划思路，但是不同的产品，不同的平台的业务需求不同，对权限系统的需求度也不一样。在设计时，根据自己的业务特点进行筛减、整合。

**二、角色权限系统实例分析**

     目前大多数B端产品都会建立比较复杂的角色分类，从而满足自身平台业务所需。在B端产品的角色权限系统中，每个用户至少扮演一个角色，每个角色至少具备一种权限；两个完全不同的角色可以分配完全相同的权限；角色之间、用户之间、权限之间可能都存从属关系和并列关系。用户和角色之间是多对多的关系，角色和权限也是多对多的关系。所以说理解RBAC模型只是基础，重要的还要结合具体业务进行设计，下面将介绍三个项目实例来说明用户、角色、权限之间的变化关系。

**2.1 某工程类项目管理平台。（由于项目本身是内部软件运行，所以会回避一些关键词）**

**项目背景说明：**

      整个平台是提供一套完整的工程项目现场管理的解决方案。

* 权限划分方式多样，变化性强：在平台，上一个公司可以同时承接多个项目，一个项目内同时又有多个公司，而这些公司之间又没有必然的联系。公司甲和公司乙在项目A中存在上下级关系，而在项目B中可能又是同级关系。
* 角色类型多样：在整个平台中，角色类型既有平台级的，同时也有公司级的，项目级。在具体的业务事件上，又有不同的角色类型。

**解决方案：**

       通过上面的描述，我们简单有一个初步认知，下面我们针对业务特点，我们做简要分析，如“图2.1 项目管理平台角色用户分析表”所示。

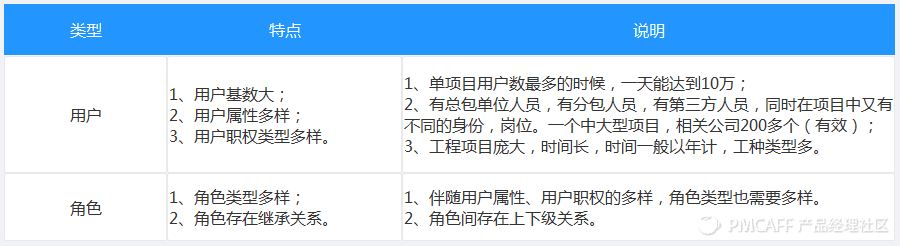
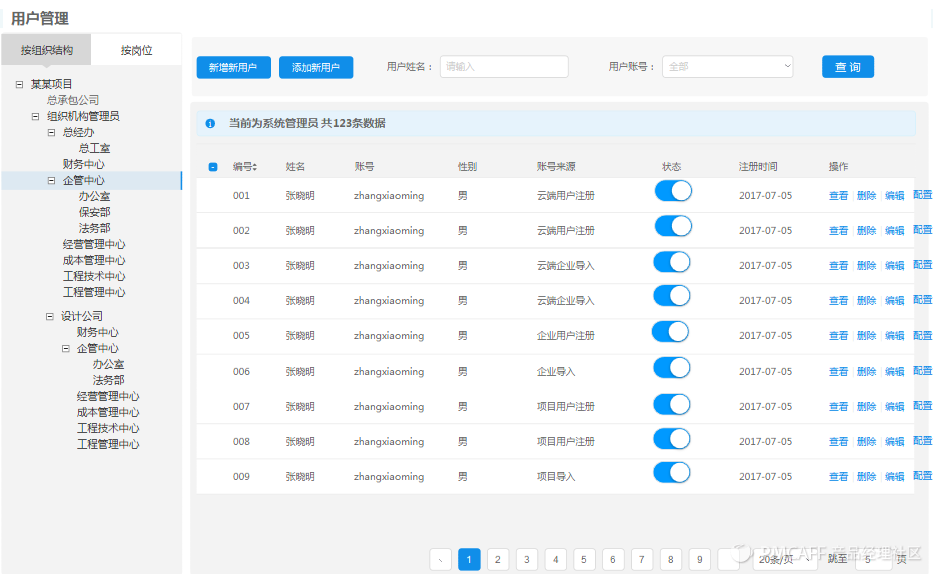


                                          图2.1 项目管理平台角色用户分析表

如图中所写的那样，我们在设计权限系统时候，要更加注重系统的扩展性，兼容性。在具体设计时，将平台的角色权限系统分为系统级权限系统、企业级权限系统、项目级权限系统、用户级权限系统。接下来具体说明下项目级角色权限系统。其他子系统逻辑相似。

**用户管理**

**说明：**

     1、将项目中，各个公司人员分开管理，构成整个项目的用户表；

     2、此处仅仅是对用户信息的增删改查；

     3、**按照组织机构、岗位将用户进行分类。让用户拥有现实中的岗位信息和组织机构信息。**

     4、可查看用户信息。用户信息分为账号信息、登录信息、个人基本信息、职位信息。

**角色**

       该平台中的角色权限系统的关键点就是如何将角色定义好， 角色是权限集合和用户集合的综合体， 而在实际工作中，一个岗位，一个组织机构都有自己的规定的工作内容，不同的工作内容对应不用权限，**所以将用户现实中拥有的身份（岗位、组织机构）当作角色的一种类型，这样就很自然的将用户实际工作与角色权限系统整合在一起**。

      但是，用这种逻辑设计角色权限系统时，应满足两个条件。第一、公司组织机构职权和权限划分相同。（说明：一个软件开发的公司引入了一个项目管理软件，这时候就不能直接用组织机构当角色，因为在组织机构中会计这种岗位在项目管理软件上没有权限。但是如果是引入一个OA软件，就可以直接可以把组织机构当成角色的一种）；第二、岗位与岗位之间、组织机构与组织机构之间有明显的职权区分。因为职权太过于相似，也没有必要用组织机构做角色，这样反而会增加使用负担。（说明：例如企业微信除了一些管理员外，大多数都是普通用户）

     将组织机构当成一种角色，给部门配权限，部门中的人就用于了相应权限。操作页面如“图2.2组织机构配权限”所示（图中是组织机构，岗位类似）



                                                        图2.2组织机构配权限

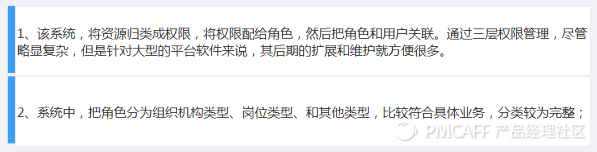
看到这里有人就有疑问，给部门整体配权限是不是部门负责人和员工一样的权限，在权限上就没有区分度了。**其实这也是该平台设计的特色。除了组织机构配权外，还有岗位配权，作为具体一个部门的部门经理就可以配适合他的权限。而在组织机构出，只配置适合整个部门的权限。这样做灵活性就高了很多，方面后期调整扩展。**另外，如果要实现某些具体的上下级权限关系也不仅仅都要通过权限来实现，该平台的一些业务上使用的工作流引擎，指定不同的角色就可以实现任务的流转，完全可以实现不同人做不同权限的事，同时也避免角色之间的互斥（一个人既是申报人又是审批人）。还有很多其他业务设计流程，与角色权限系统相辅相成共同实现平台运作。

除了这种具体身份的角色外，平台还有系统角色，比如，系统安全管理员，项目超级管理员等等，这些角色都可以直接配给用户。

**权限**

前面也说过，权限是资源的集合。该系统的做法是将所有应用的页面、模块、功能、字段都作为一种资源类型，进行统一权限管理。详细页面就不贴了，这里说明下，由于改平台需要进行权限管理的应用模块多，所以在将权限配个角色前，先将权限做了分类，增加了一层配置。也就是说，将资源构成一些列的小权限，然后在将小权限合成大权限，形成一个权限树结果，然后，不同角色从树上找适合自己的权限。

**总结：**



该平台中，角色权限系统关系图如图2.3所示

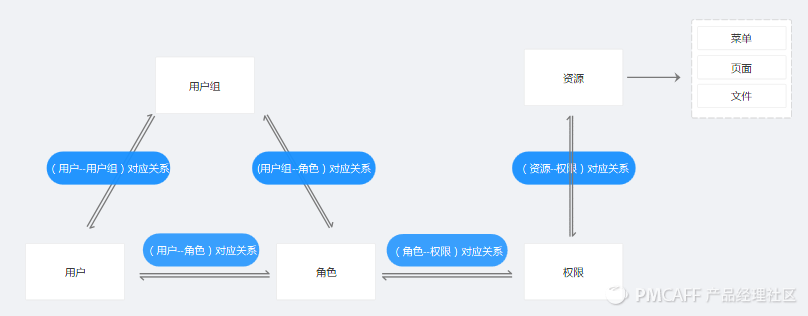


                                            图2.3工程管理项目用户-角色-权限关系图

**2.2 禅道**

**背景说明：**

禅道是一个可以进行项目管理、产品管理、文件管理的软件。接下来就简单分析下禅道的角色权限系统。

**解决方案：**

禅道是一个项目管理软件，对于一个软件公司，也只有和软件相关的人会使用，所以简单分析下禅道用户和角色的特点，如“图2.4禅道角色用户分析表”所示

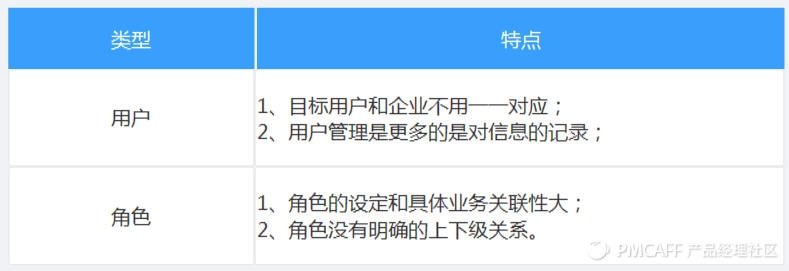


                                                       图2.4禅道角色用户分析表

**用户：**

禅道是一个项目管理 软件，所以平台上的用户更多的是**项目相关的用户**，和企业自身的组织机构，没有一一对应关系。并且此处的用户只需要做用户信息的基础管理即可。

用户列表图



                                                          图2.5禅道用户列表

**角色：**

**项目定位清晰，业务流程明确，有自身完整一套规范的使用流程，所以在角色设定上，主要是按目的进行角色设置。**

用户属性，按照目的将用户分成不同的用户组，代表一种角色。

权限属性，根据业务针对每一种角色进行配置。

禅道的角色权限目的明确，软件本身结构简单，都做在一个页面上。

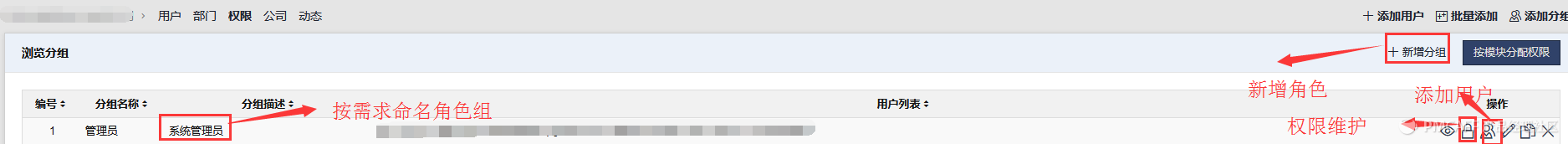


                                                        图2.6禅道角色权限管理表

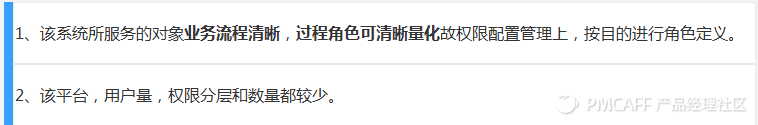
**权限：**

该系统权限数量不是太多，所以直接将资源配置在角色上。



                                                       图2.7禅道权限配置表

**总结：**



该平台，把用户组管理和权限管理整合在一起。关系图如下所示；

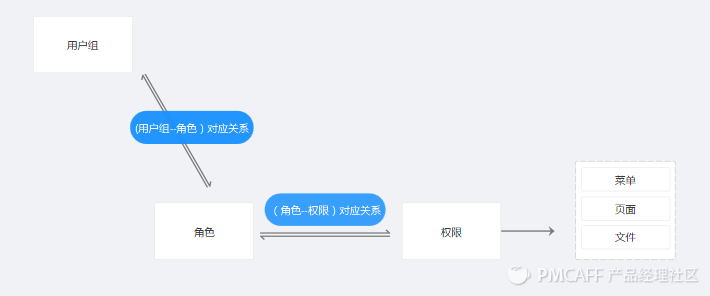


                                                   图2.7禅道用户角色权限关系图

**2.3 某某CRM系统**

此系统中，对角色划分需求度更低，只需要区分出管理员和普通用户即可，没有复杂的用户关系。



                                                     图2.8某系统权限配置图

该系统，**由于角色类型说，软件本身是辅助性软件，所以直接对用户进行权限配置。**关系图如下所示；

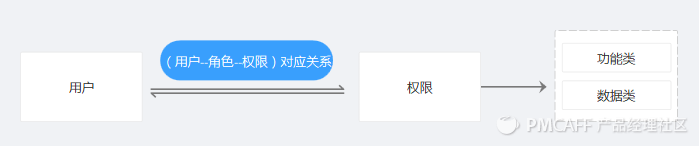


                                     图2.9某系统禅道用户角色权限关系图

**三、设计角色权限系统时的几条建议**

上面列举的三个角色权限实例都是基于RBAC模型。三个案例由繁至简，结合不同的业务需求，进行模式调整。尽管样式不同，但万变不离其宗，理解清楚RBAC模型后，结合自己的业务就可以设计出一套符合自己平台需求的角色权限系统。

上面三个案例比较有代表性，可以结合参考。但同时，自己设计时一定要先理清“2个关系3个多少”

1、用户与用户之间的关系。用户有哪些？他们的身份是什么？他们的关系又是怎样？他们做的事有什么不同？用户之间有哪些不同的属性；

2、用户与角色之间的关系。是怎样的对应关系，业务上需要哪类角色？角色间有什么差异？角色间是否有交叉和从属关系？

4、用户量多少。

5、角色类型多少。

6、需要进行权限控制的应用有多少。哪些页面，哪些功能，哪些数据需要进行权限控制。

**后续**

以上是自己主导过，或者使用过的系统平台。根据自己的理解给出的一套解读方案。有不足之处，希望大家多多交流。

自己在工作中有些许总结。最近准备将零散的总结整合到一起，分享出来，大家一起讨论下，互相沟通。主要分为，设计技能、业务技能、思考分享、其他几个维度。

