基于 LDAP 和 RBAC 的文档管理系统①

谢志新, 肖炳甲, 黄 静, 王 枫

(中科院等离子体物理研究所 计算机应用研究室, 合肥 230031)

摘 要: 随着 EAST 大科学工程的进行, 产生了越来越多的项目文档, 为了更有效的管理海量文档资料, 文档共享, 需要设计一个兼具丰富文档管理功能、强大用户管理功能以及完善权限控制功能的集成文档管理系统. 通过 PHP 和 MySQL 设计控制逻辑和数据结构, 实现文档的创建、修改、删除、查找、上传和下载等功能, 采用轻量目录访问协议实现统一用户管理, 采用基于角色的访问控制技术实现高级权限分配, 并根据用户角色信息实现文档版本控制和工作流管理, 最终满足 EAST 文档管理的需要.

关键词: 文档管理系统; 轻量目录访问协议; 基于角色访问控制; 版本控制; 工作流

LDAP and RBAC Based Document Management System

XIE Zhi-Xin, XIAO Bing-Jia, HUANG Jing, WANG Feng

(Division of Computer Applications, Institute of Plasma Physics, Chinese Academy of Sciences, Heifei 230031, China)

Abstract: With the advancement of the EAST project, more and more documents are produced. In order to effectively manage huge amounts of documents, share documents, and promote the collaboration with other organizations, we need to design an integrated document management system, which has a powerful document management function, a strong user management function and a blameless permission control function. Using PHP and MySQL to design control and data structure, the function of document create, edit, delete, search, upload and download is accomplished. The Lightweight Directory Access Protocol technology is used to manage users and achieve access control. By using the Role Based Access Control method, the advanced user permission distribution is achieved. Combining with user and role information, document version control and workflow management are also implemented. Finally this system meets the demand of EAST document management.

Key words: document management system; lightweight directory access protocol; role based access control; version control; workflow

随着互联网的飞速发展, 越来越多的企事业机构通过网络开展科研协作及各类互联网业务, 作为信息流通主要载体的电子文档呈爆炸式增长, 给传统的文档管理带来巨大的冲击, 如何快速有效安全的存储、管理和共享文档, 并控制大量用户对文档的访问和操作权限, 成为新型文档管理的重点设计需求[1].

国家大科学工程项目全超导托卡马克实验装置 EAST, 是我国独立自主发展的具有大拉长比的非圆截面等离子体位形的稳态聚变装置,它的目标是针对目前建造托卡马克核聚变堆尚存的前沿性物理问题展开探索性的实验研究,从而使我国磁约束核聚变研究进入世界先进行列[2]. 项目运行过程中产生了海量的项目、实验、会议等各类文档, 且由于各研究组织间的位置分散性、管理独立性, 这些文档分布存储在不同的计算机上, 出现管理混乱、文档冗余、版本有效性和权威性的普遍缺失、缺少权限控制管理等问题. 为了解决这些问题, 需要一个集中的文档管理系统.

目前虽然一些商业或免费的文档管理系统, 如

① 基金项目:国家磁约束核聚变能发展专项(2012GB105000,2011GB101000)
收稿时间:2014-01-07;收到修改稿时间:2014-03-03

eDoc、DocCare、易度等, 也具有一定的文档管理功能, 但仍然缺乏灵活的按需定制、良好的开源支持等问题. 同时考虑到 EAST 大多数服务器采用 Linux 系统, 许多服务程序也充分利用了开源的资源; 参与 EAST 项目的研究人员数量庞大, 国内外合作交流组织众多; 并且 EAST 项目涉及多个不同应用系统, 为了使多个应用共享用户模块, 需要一个统一的用户管理模块, 只需设定用户对不同应用系统的访问权限, 即可实现用户模块的一次建立重复使用. 而这些都是一般文档管理系统无法实现的, 因此我们需要开发一个特定的文档管理系统 EAST Document Management (EDM).

基于此本文在深入研究传统文档管理系统的基础上, 利用轻量目录访问协议和基于角色权限管理的方法, 提出了一种具有复杂权限控制功能的基于 Web 的多用户文档管理系统解决方案.

1 系统总体架构

1.1 系统功能架构

系统按逻辑功能来分主要包括两个部分, 文档部分和用户部分, 如图 1 所示,

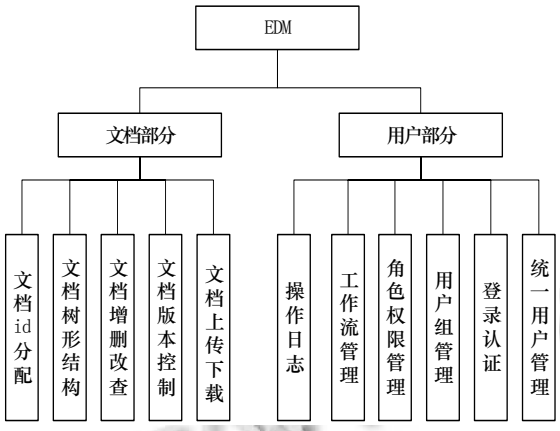

图 1 系统功能架构

文档(包括文件和文件夹)部分主要用于完成海量文档资料的安全集中存储管理. 为了唯一定位和高速检索文档为文档设计标识 id; 为使文档结构层次分明采用树型结构组织显示文档; 同时包括高速文档的上传下载、完善的文档版本控制等功能. 用户部分首先包括统一的用户管理, EDM 采用 LDAP 目录服务来存储和管理用户信息, 实现多个应用系统以不同的方式共享 LADP 用户信息, 同时由于 LDAP 特殊的数据结构, 可以实现按企业部门组织结构存储用户信息. 完善的登录认证, 只有 LDAP 中被授权的用户才能访问系统. 为批量设置用户权限和修改用户信息, 采用用户分组管理. 采用 Role Based Access Control (RBAC) 设置文档角色信息, 为角色赋予权限, 从而使扮演不同角色的用户或用户组成员对相应文档拥有不同的操作权限. 为实现文档编辑、提交、评阅、审批等办公流程的自动化管理, 开发设计文档工作流管理功能. 系统记录用户对文档执行的操作历史到操作日志中, 保证系统安全, 并能在发生错误时快速找出发生错误的地点, 有利于系统维护.

1.2 系统设计架构

EDM 是一个基于 Linux+Apache+MySQL+PHP (LAMP)设计平台的 Web 应用[3]. 采用 MVC(逻辑层、表现层、控制层)网站架构模式, 将应用程序的逻辑层和表现层进行分离, 允许网页只包含很少的脚本. Model(逻辑层)专注于数据库的各类操作, 代表数据结构; View(表现层)是展示给用户的信息, 可将 Model 层传过来的数据根据需要定制各种丰富美观的界面元素; Controller(控制器层)是协调控制 Model、View 及处理 Http 请求的资源中介. MVC 并非唯一的设计模式, 但它拥有比其他设计模式更高的可重用性、模块独立性、灵活可配置性, 因此系统考虑采用 MVC 软件架构模式.

后台采用 CodeIgniter 作为 PHP 开发框架[4], 设计控制逻辑, 处理与数据库的连接和各类数据操作. 采用 Eclipse 插件 ERMaster 设计数据表, 并根据表之间的关系建立 E-R 图. 前台使用 jQuery 作为主要的界面开发库, 同时为实现与用户的动态交互、部分刷新页面、减小服务器负担, 采用 Ajax 异步交互技术[5].

2 文档管理

为了唯一标示 EDM 中的文档, 便于定位和检索文档, 需要为每个文档分配配一个唯一的标示 id. 标示 id 相互之间绝对不会重复, 且能够容纳足够多的文档, 同时 id 要求有较强的随机性和易记性. id 由 6 个字符组成, 每个字符的取值范围是 26 个英文大写字母和 0~9 数字, 例如 id=FH6N3E, 所以总共可容纳 366≈2.2 亿个文档, 足够满足文档容量需求. 生成 id 的方法有很多种, 主要可划分为两类: 预先生成和在线生成. 预先生成是预先按一定算法生成所有 id 值, 每当创建

新的文档时从中获取一个未使用过的 id. 在线生成是为每一个用户按照算法在线生成一个不重复的预留 id, 用户使用自己的预留 id 创建文档. 两种方式都能保证 id 的唯一性, 但前者 id 的存储空间与所有可能的 id 值个数成比例, 后者只与用户数量成比例, 节约存储空间, 因此 EDM 采用在线生成的方式创建 id.

为满足用户使用文档管理系统的惯性操作, EDM 采用左树右表的形式组织文档, 即网页左边显示文档目录树, 如果用户知道文件的存储路径, 可以直接根据该目录树找到文档, 右边以表格形式显示用户当前所在目录的所有子文档. 文档的实际存储与文档的显示分离, 为了实现文档在网页中的树形显示, 需要设计相应的 MySQL 数据表来完成. 目录树中的每一个文档作为树的一个节点, 在树中对文档进行增删改查等操作, 然而 MySQL 是典型的以二维表形式存储记录的关系型数据库, 不能将目录树结构直接存入, 因此必须为文档节点设计合理的数据结构, 节点要拥有足够的属性信息来支持这些功能. EDM 的文档树表结构设计为{id, title, pid, path},这样直接记录了节点之间的继承关系, 非常直观, 并且很方便地支持目录扩展, 每个目录可以根据需要无限极向下扩展.

为了实现对文档的创建、修改、删除、上传、下载等各类操作, EDM 主要借助 PHP 开发框架 CodeIgniter 的文件辅助函数, 下载辅助函数, 和文件上传类等来完成; 同时使用 PHP 数据库操作函数更改 MySQL 数据库相应数据表的内容, 文档操作与表操作一致, 实现文档显示与文档实体同步.

3　用户权限管理

对于同时拥有用户和对象的系统, 除了丰富的对象管理功能之外, 设计严谨的用户访问控制, 并为用户分配合理的操作权限, 确保用户对相应的对象能且只能执行被授权的操作, 是一个完善和个性化系统必不可少的设计重点. 文档作为 EDM 系统中用户的主要操作对象, 考虑到文档包含文件和文件夹, 并且是分级按路径存储, 需要设计更复杂的用户权限管理模型. EDM 采用 LDAP 统一管理用户实现访问控制, 结合 RBAC 实现完善的权限分配功能.

3.1 LDAP 统一用户管理

Lightweight Directory Access Protocol (LDAP)轻量目录访问协议, 是在 X.500 标准的基础上做了简化, 使实施起来更加宽松简单, 且可以根据需要定制功能, 运行在 TCP/IP 协议上, 更好地支持了 Internet 的访问需要. LDAP 协议常用来发布目录信息到许多不同资源, 作为一个集中的地址本使用[6].

LDAP 以特定的数据结构存储目录信息, 是一种特殊的数据库. 但是和一般的数据库不同的是, 它对查询操作进行了优化设计, 读取速度比一般数据库要快很多, 因此用 LDAP 统一管理用户信息, 还能加快用户登录系统时的认证过程. LDAP 定义了目录的组织结构和信息单元, 以树形的层次结构来存储数据, 当用 LDAP 存储某个企业的部门、工作组、员工信息时, 树形层次结构能直观明了的反应企业内部结构, 同时还可以根据目录树的结构给予某一子树下的所有结点统一设定权限, 便于管理. LDAP 的信息单元称为条目, 目录树中的每个结点就是一个条目, 每个条目有唯一的分辨名 DN [7].

EDM 中的 LDAP 目录结构如图 2 所示, 根目录的分辨名是“dc=test,dc=com”, Test 子目录是单位内部组织, Others 是外部合作组织. D01～D013 是根据实验室将内部成员分组, 每个实验室的 people 子目录记录该实验室的成员信息. 成员信息主要包括唯一的分别名、用户名、性别、邮箱等.

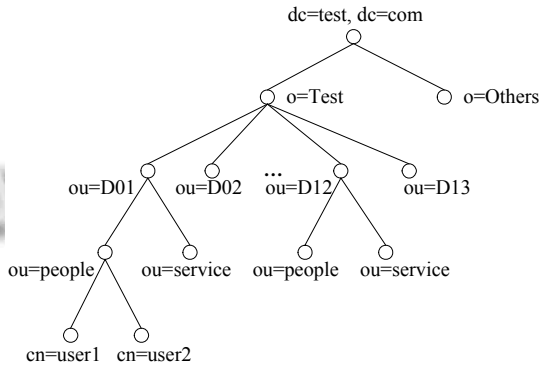

图 2　LDAP 目录结构

为了对用户登录 EDM 进行访问控制, 在 LDAP 数据库的用户信息中增加了信任方式 trustModel 和授权服务 accessTo 两个属性. 用户登录时, 除了需要正确的用户名和密码之外, 还必须能够通过 LDAP 的访问过滤规则, 即如果用户的信任方式是"fullaccess”,则用户有权访问 EAST 项目所有应用系统, 如果用户的信任方式是“byservice”, 则用户只能够访问授权服务属性 accessTo 中对应的应用系统. 过滤语法如:

filter=(&(uid=username)(|(trustModel=fullaccess)(&(trustModel=byservice)(accessTo=edm)))), 如此实现了用户管理和访问控制.

3.2 基于角色权限管理

EDM 采用基于角色的访问控制技术 RBAC, 与一般访问控制技术相比, 基于角色的访问控制技术对系统操作的各种权限不是直接授予具体的用户, 而是在用户集合与权限集合之间建立一个角色集合. 每一种角色对应一组相应的权限. 一旦用户被分配了适当的角色后, 该用户就拥有此角色的所有操作权限[8]. 这样做的好处是, 不必在每次创建用户时都进行分配权限的操作, 只要分配用户相应的角色即可, 而且角色的权限变更比用户的权限变更要少得多, 这样将简化用户的权限管理, 减少系统的开销.

按照 RBAC 的设计思想, 首先根据实际情况为文档定义各种角色信息, 因为文档包括文件夹和文件, 它们对应的角色不同, 角色又有级别之分, 因此文件夹的角色根据级别从高到低包括文件夹的负责人、添加者、浏览者; 文件的角色根据级别从高到低包括作者、合作者、审批者、评阅人. 其次确定每个角色拥有的权利, 即操作权限, 角色的权限根据角色级别, 拥有向下包含的关系, 即高级别的角色拥有级别低于它的角色的所有操作权限. 最后为文档的各个角色分配相应的用户或用户组. 如此该用户或用户组对该文档便可执行对应角色所拥有的操作权限. EDM 用户信息都存储在 LDAP 数据库中, 而这些角色、权限、用户组、文档等相关信息都保存在 MySQL 数据库中. 系统通过两个数据库实现复杂权限管理, 并保证 LDAP 和 MySQL 两个数据库的同步. 为实现上述 RBAC 权限分配模型, 涉及到部分 MySQL 数据表, 它们的实体关系模型如图 3 所示.

其中 user 表可以作为对 LDAP 中用户的扩展, 以提高 EDM 系统的可移植性, 当企业或组织没有 LDAP 目录服务时, 可用 user 表建立自己的用户系统. groups 表存储组相关信息, 主要字段有{group_id, name, type, create_user……}, 组与用户之间是多对多的映射关系. 根据用户组的来源将 groups 中的 type 分为三种, 即 AD Group (AD)、Global Group (GG)、Local Group (LG), AD 组来自于 LDAP 服务器, 只能由 LDAP 管理者创建; GG 组由 EDM 系统管理员创建, 在 EDM 全局范围内有效; LG 组由文件夹的负责人创建, 只在当前文件夹以下有效. dir_group_role 是文件夹角色分配表, 文件夹以用户组为单位分配角色, 文件夹默认继承上一级目录的权限设置, 文件夹的负责人也可以根据需要修改权限设置. doc_user_role 是文件角色分配表, 文件以用户为单位分配角色, 文件的创建者即文件的作者, 可以设置文件的合作者、评阅者、审批者. 没有分配角色信息的用户将不能对文档进行访问和执行任何操作.

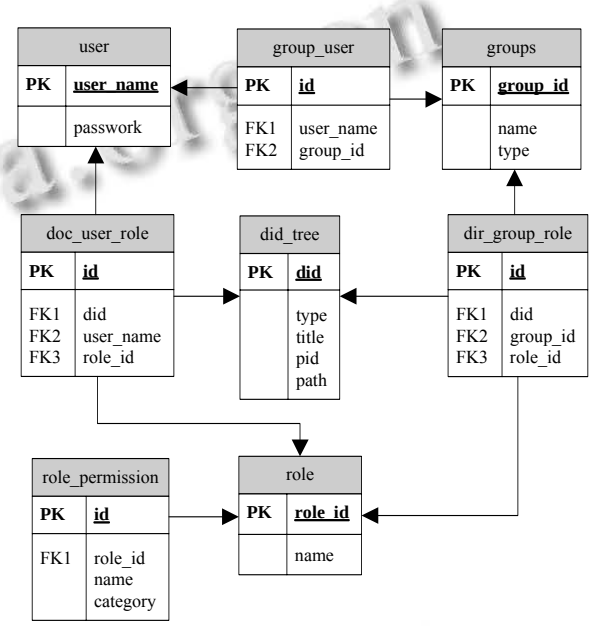

图 3 RBAC 关系数据库 E-R 图

3.3 版本控制和工作流

大部分文档都是通过多次编辑、重复修改生成, 特别是对于由多人参与的项目文档, 文档的更新迭代频率更高, 为记录文档的迭代过程, 比较每次修改的不同, 设计在每次文档做大的变动之后都可根据需要将其作为文档的一个新的版本保存下来, 每一个版本分配一个按一定规则排列的版本号 version_id. 用户通过文档唯一标示 id 查看文档时, 默认访问的是该文档的当前版本, 也可以通过文档 id 和 version_id 直接查看文档某一特定版本. 对于每个文档, 都有且只有一个当前版本. 系统根据文档路径、版本信息和文件名存储实际的文档内容. 文档所有版本的一些公用信息及当前版本的基本信息存储在 MySQL 的 doc_basic 表, 如文档标题、作者、创建日期等; 所有历史版本的详细信息存储在 doc_version 表.

系统中的文档从作者编辑、提交、评阅者评阅、

审批者审批, 是一个从时间和空间上离散, 并且需要多人参与完成的业务流程, 为了方便办公, 节约时间, 提供更高效的工作效率, 引入文档工作流功能. 将文档这一过程组织成一种预定的逻辑规则以恰当的模型进行表示并对其实施计算[9]. 利用之前定义的文档角色信息, 采用基于状态的工作流控制技术, 能够快速、高效的实现文档工作流程管理.

EDM 中文档工作流的具体实现, 主要依据文档基本信息 doc_basic 表中的 status 字段, 该字段表示文档所处的工作状态, 总共有四种工作状态, 分别是编辑中(In work)、提交(Signed)、审批通过(Approved)和不通过(Disapproved), 这四种状态按流程先后分为三个级别, 流程越靠后级别越高. 文档从创建编辑开始到最后审批, 会邮件通知相应角色去执行工作流程中的相应操作, 工作状态也会跟着流程的执行发生相应的变化[10].

不考虑版本, 文档工作流是一个单通道线性流程; 而 EDM 文档是分版本存储, 每个版本都有自己的一个线性工作流, 即 doc_version 表中也设有状态字段 status, 整个文档的工作流类似一个多通道并行推进的流程, 文档的最终工作流状态取决于当前版本. 因为文档有且只有一个当前版本, 当前版本的确定取决于工作流状态优先级算法, 该算法主要是根据状态级别和版本大小设计的, 具体算法模型如图 4 所示.

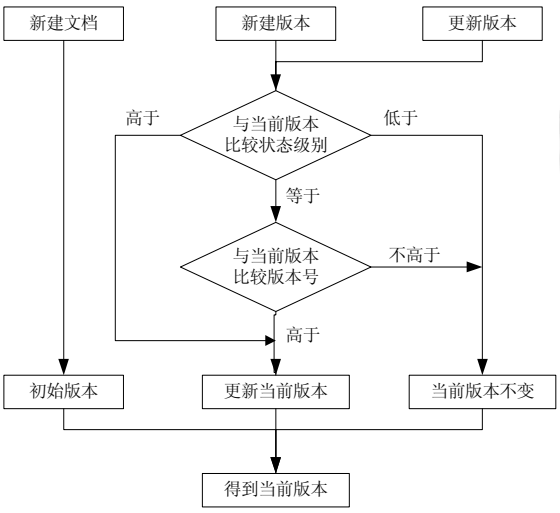

图 4 工作流优先级算法模型图

如果新建文档, 则默认这是文档的第一个版本, 也是当前版本, 文档的工作流状态即为这个初始版本的状态. 如果文档已经存在, 当为文档新建一个版本或更新某个版本时, 将新建或更新版本的工作流状态级别与当前版本的工作流状态比较, 若低于, 则说明当前版本是更成熟和被他人接受的版本, 当前版本不变; 若高于, 则说明新建或更新版本是更成熟和被他人接受的版本, 当前版本改变; 若等于, 则将新建或更新版本的版本号与当前版本比较, 版本号高者作为当前版本.

除了邮件通知文档相应角色去完成工作流中的相关操作外, 用户登录系统之后, 还可以通过个人工作台, 查看工作流中未完成的任务, 了解最近用户需要对系统中那些文档进行何种操作. 个人工作台还显示系统最新更新的文档、用户最近执行过的操作、个人收藏夹、个人订阅、用户日程等信息. 这些信息存储在后台 MySQL 数据库的相应表中, 前台负责将查询结果显示给用户浏览.

4 结语

本文主要研究基于 LDAP 和 RBAC 的文档管理系统的设计与实现. 在基本的文档管理基础上, 系统重点设计实现如何将 LDAP 统一用户管理和基于角色的访问控制技术相结合, 实现系统从用户统一管理、登录认证、权限设置、到角色分配整个用户权限分配模型的设计; 完成版本控制、工作流管理和邮件任务通知等功能. 我们采用模拟数据对 EDM 系统进行了负载测试, 在单台 DELL R720 服务器上, 容纳 1000 个用户, 100 万个文档, 1000 万个版本, 1 亿个附件的情况下, 系统可以流畅的运行和操作, 基本实现了设计要求. 在将来会对系统做进一步完善, 如增添文档推荐功能、文档模板化功能、移动应用等.

参考文献

1 熊铭,吴梅,薛小平,阮永良.基于 Java Web 组件技术的软件项目文档管理系统.计算机工程,2002,28(12):257–259.
2 胡立群,张晓东,姚若河.EAST 托卡马克的中性束注入方案.核技术,2006,29(2):149–152.
3 肖维明.基于 PHP + MySQL 的网站开发.物流工程与管理,2009,31(6):90–92.
4 顾大刚.基于 MVC 模式的 PHP 开发框架 CodeIgniter.江西科学刊,2009,27(5):722–724,203.
5 Holdener AT.Ajax 权威指南.陈宗斌译.北京:机械工业出版社,2009.
6 常潘,沈富可.基于 LDAP 的校园网统一身份认证的实现.计算机工程,2007,33(5):281–282.
7 徐鹏,王耀.Moodle 及邮件系统基于 LDAP 统一身份认证.计算机应用与软件,2011,28(12):232–235,288.
8 许峰,赖海光,黄皓,谢立.向服务的角色访问控制技术研究.计算机学报,2005,28(4):686–693.
9 阳万安,李彦.通用版本控制系统的研究和设计.计算机工程,2008,34(12):283–285.
10 谢子松,武友新,牛德雄,江恭和,张烨.于 UML 的工作流建模的研究与应用.计算机系统应用,2008,14(2):22–25.