

构件仓库的研究与应用^①

韩荣振¹, 景新海², 王 刚¹

¹(山东师范大学 信息科学与工程学院, 济南 250014)

²(中创软件股份有限公司, 济南 250014)

摘 要: 为了解决构件的使用和管理问题, 提出构件仓库的解决方案。首先, 研究了构件仓库的总体结构和逻辑结构, 从整体上分析构件仓库。其次, 介绍了构件仓库的部署。最后, 从构件使用、构件发布、构件管理和仓库设置四个方面研究了构件仓库的应用。对构件仓库的使用, 根据用户角色的不同分别赋予不同的权限, 合理地管理构件仓库中的构件。采用构件仓库管理构件, 可以有效地提高软件开发的效率, 使构件能够更合理地得到复用, 提高构件的复用率。实践表明, 构件仓库能够有效地解决构件的使用和管理问题, 为软件开发提供有力支持。

关键词: 构件仓库; 构件; 软件复用; 构件管理; 软件工程

Research and Application of Components Repository

HAN Rong-Zhen¹, JING Xin-Hai², WANG Gang¹

¹(Department of Information Science and Engineering, Shandong Normal University, Jinan 250014, China)

²(CVIC Software Engineering Co. Ltd., Jinan 250014, China)

Abstract: To solve the problem of application and administration of components, this paper puts forward a solution to the components repository. Firstly, the paper studied overall structure and logic structure, gave whole analysis to the components repository. Secondly, it introduced disposition of the components repository. Lastly, this paper studied the application of components repository in components using components release, components management and repository establishment. To the uses of components repository, it will give different permissions according to the different roles to manage the components in the components repository. Using the components management in the components repository, it can improve the efficiency of software development effectively, reuse the components more reasonably, and improve the reuse rate. Practices show that the components repository can solve the problem of administration to the components, and give strong support to the software development.

Key words: components repository ; component; software reuse; component management; software engineering

经过长期的发展, 软件工程的思想和方法已得到了广泛的应用, 随着各种软件开发方法的普及和改进, 软件生产的效率逐步提高, 软件的质量也得到了改进。但是相对于社会对软件的需求和软件危机对软件工程的影响, 软件工程还有很大的发展潜力。近几年来, 人们开始认识到, 要真正实现软件的工业化生产方式, 保证软件生产的高效率和高质量, 软件复用是一条切实可行的途径^[1]。当前, 软件构件技术已成为研究的

热点, 软件构件是指软件系统中具有相对独立功能、可以明确辨识、接口由契约指定和语境有明显依赖关系、可独立部署、可组装的软件实体^[2]。它由接口、实现和部署三大要素构成^[3], 其研究内容包括构件标准和模型、构件的生产和获取、构件的规约和描述、构件的分类和组织、构件的检索和组装、基于构件的价格分析, 以及软件体系结构、软件复用支持工具和管理手段、基于复用的软件开发过程等方面。研究工

① 收稿时间:2011-10-21;收到修改稿时间:2011-11-17

作有很大进展和众多成果，而且还出现了一些产业构件标准，如 Microsoft 的 OLE - COM、OMG 的 CORBA/OM 等。这标志着软件复用已进入蓬勃发展的时期^[4]。

软件构件经过近几年的发展，已经产生了各种各样的软件构件，构件的功能、应用领域、适应范围等各不相同，在软件开发中很难找到合适的构件，影响了软件复用，降低了开发的效率，因此对于构件的管理就提出了更高的要求。本文介绍一种构件仓库的技术，可以更好地管理构件，在软件开发过程中，更快地找到软件所需要的合适的构件并应用到软件中，有效地提高软件开发的效率。

1 构件仓库的结构

构件仓库是构件软件工程的基础设施，基于 maven^[5]的基本原理,中创软件根据构件仓库应用的级别分为资源仓库、企业级仓库、项目级仓库。资源仓库是指不仅仅存储构件，还存储其它资源，如文档。企业级仓库即公司级仓库、中央仓库，是整个企业的核心仓库。项目级仓库是指比企业级仓库低一个层次；一个子公司的构件仓库也可视为一个项目级仓库，一个项目级仓库可以包含一个或多个项目。它把基础构件和业务构件积累起来，可以在企业应用中复用和锤炼，以获得更高效的开发、更稳定的质量和更高的性能。构件仓库提供了对构件存储、检索、共享、反馈的操作与管理。通过项目级仓库与企业级仓库，为团队开发提供版本一致性和依赖管理。最终形成一套完整构件管理机制，使软件知识和技术成果的沉淀可以通过构件库的形式得以实现和发展。

1.1 构件仓库总体结构

本文研究的构件仓库整体上分为功能层、容器层、数据层。功能层为用户提供服务接口，基于角色的不同，提供不同的服务。容器层解决构件仓库在应用服务器上的部署问题，构件仓库可以部署在 tomcat、websphere 等 web 服务器上，也可以作为独立的服务进行安装使用。在容器层部署成功后，通过互联网/局域网访问系统，也可以通过 Web Service 和 WebDAV 的接口形式使用仓库提供的服务。数据层是构件仓库的持久化层，它包含三个数据库，用户数据库存储的是用户信息和权限信息，用户的创建、修改、删除都要对用户数据库进行操作；构件数据库存储的是仓库

中构件的相关信息，包括构件标识、组织标识、构件版本等；文件系统数据库存储的是构件实体、POM 文件以及构件的相关配置信息的文档，用户上传的文档也会存储在这个数据库中。构件仓库的三层设计保证了构件存储的安全性，也给用户的使用提供了便利，提高了构件检索的速度，能够有效地提高软件开发的效率和构件的复用率。图 1 构件仓库总体结构图形象地展示了构件仓库的总体设计模式。

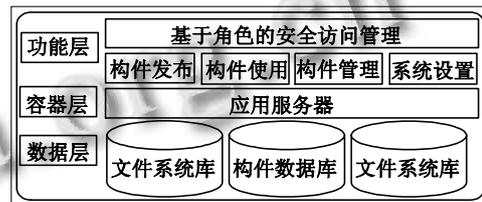


图 1 构件仓库总体结构图

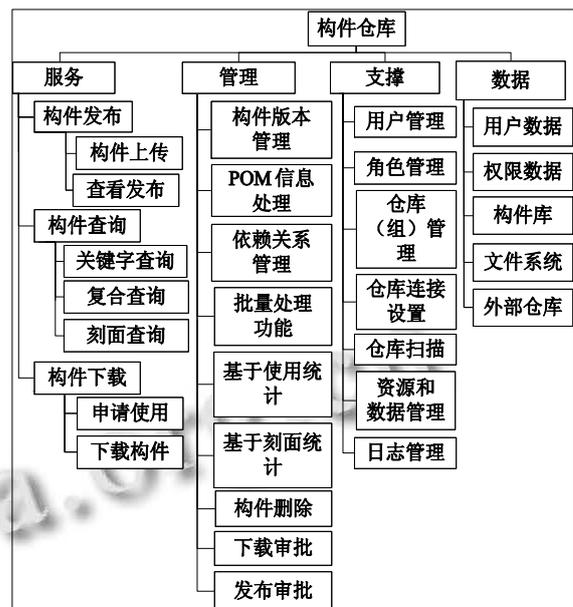


图 2 构件仓库逻辑结构图

1.2 构件仓库逻辑结构

从逻辑上，整个仓库可以分为数据层、支撑层、管理层、服务层。自下而上，数据层提供系统所有的数据存储的支撑，主要包括用户数据、权限数据、构件信息、构件实体、索引文件；数据层是整个构件仓库的基础；支撑层提供整个系统的基础数据初始化和数据准备的工作，包括最重要的如用户和角色数据；管理层提供整个系统的日常管理工作，包括如构件的属性管理；服务层给最终用户提供构件的发布、构件

查询、构件下载功能。

2 构件仓库的部署

构件仓库部署分为两部分描述，一部分是构件仓库部署结构，一部分是构件仓库网络结构。由于数据存储方式分为数据库存储和文件系统存储，如果有需要数据库存储部分可以分离出来单独部署，但文件系统存储部分不适合分开存放。

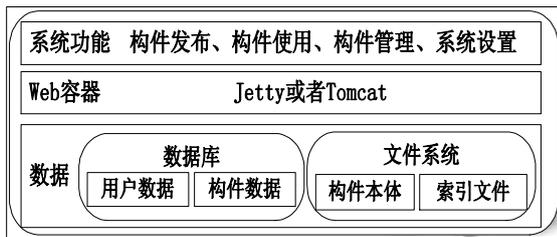


图 3 构件仓库部署架构图

构件仓库网络结构包括普通集中式部署和高级部署形式，另外还可支持集群。

其中，普通部署方式适合对性能要求不高的开发方式，如项目级仓库。图 4 展示了普通集中式部署方式的网络结构。

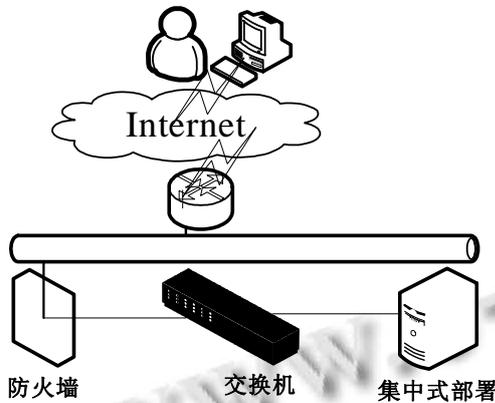


图 4 网络结构（普通集中式部署方式）

而高级部署方式适合对性能要求较高的开发方式，如企业级仓库。图 5 展示了高级部署方式的网络结构。

3 构件仓库的应用

根据构件仓库的模型，构件仓库的应用主要体现在构件使用、构件发布、构件管理和仓库设置四个方面。

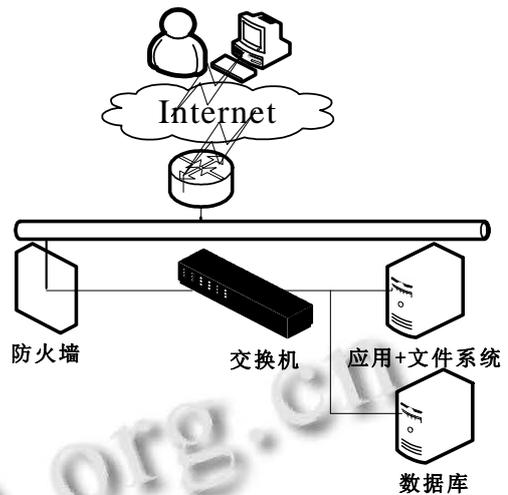


图 5 网络结构（高级部署方式）

3.1 构件使用

构件使用是构件仓库最重要的应用，主要提供构件查询、申请使用、构件下载、下载审批服务。在构件查询时，提供了多种查询方式，如基于关键字查询、基于刻面查询、复合查询。通过这些方式可以更容易的查询出与要开发的软件适合的构件，为软件开发提供有力的支持，方便构件的复用。申请使用、构件下载、下载审批等提供了对构件管理的服务，通过这些服务能够更好地控制构件的使用，规范构件使用的过程。

3.2 构件发布

构件发布包括构件上传和发布审批两个部分。构件上传是将构件存储到临时仓库，经过发布审批之后，构件将被存储到正式仓库。当构件开发完成后，构件仓库中提供了构件发布服务。通过构件上传，可以向构件仓库中添加构件。在上传构件时，需要提供构件准确的描述信息，这能更有助于构件的查询和使用，构件上传完成后，通过查看发布进一步的确认构件信息的正确性。发布审批提供管理构件发布的途径和功能，使得能更好地对构件发布过程进行管理。

3.3 构件管理

构件管理模块提供了对构件的版本、POM 信息、依赖关系、基于使用统计、基于刻面统计以及构件删除的服务。每发布一个构件，构件仓库都会相应的记录其版本、POM 信息和依赖关系。基于构件使用的统计方式是对构件的下载量、查看量的统计，并通过多种常用的图形化来展示。基于刻面的统计方式是参照刻面分类，构件能按照这些刻面进行详细的统计，并

通过多种常用的图形化进行展示。这两种统计方法给构件仓库管理员提供了构件仓库的详细信息,使管理员能够更好地管理构件仓库。

删除构件分两种情况:一是未发布的构件,此类构件不必经过申请和审批,可直接物理删除;二是已经正式发布的构件,需要首先提出删除申请,经过审批后,将构件移到仓库的缓冲区(相当于回收站),对于删除到缓冲区里面的构件,单独进行管理。

3.4 仓库设置

仓库设置包括基本的仓库设置、仓库(组)管理、日志管理、仓库扫描四个部分。基本仓库设置提供了对用户权限的管理,权限管理包括用户和角色的管理,完成用户授权功能;用户管理实现用户的创建、修改、删除、查看详细、查询、授权,通过授权在用户登录后根据授权内容显示用户所拥有的不同功能;角色管理实现角色的创建、修改、删除、查看详细、查询、角色与功能的授权。通过角色管理,系统不必给每个用户授予功能,而只需要给用户授予角色,就可以拥有该角色的功能。

仓库(组)管理是为了完成仓库功能配置功能。仓库组是用来合并多个本地仓库和代理仓库,本地仓库是远程仓库的一个缓冲和子集,当构建项目的时候,首先会从本地仓库查找资源,如果没有,那么会从远程仓库下载到本地仓库。这样在下次使用的时候就不需要从远程下载了。本地仓库可作为构件缓存服务器或者企业内部协作的交换及存档仓库。代理仓库是对远程仓库的代理,它们被用来代理远程的公共仓库,如企业级仓库、项目级仓库、子公司的项目仓库,或者其他开源组织的公共库等等。

日志管理提供了系统操作日志的查询审计功能,能详细且完整的查询系统的增加、删除、修改、查询操作。日志的详细信息包含:操作时间、操作对象、操作者、操作者机器及物理位置信息、执行的操作。

仓库扫描通过设置构件类型、自动删除类型等来实现系统对整个仓库文件的扫描,实现对文件的直接管理。通过资源和数据管理,达到仓库扫描使得文件信息和数据库中信息同步目标。通过仓库的连接设置,可以设置仓库的代理服务器。

目前,构件仓库已经在中创软件成功应用,与传统的构件数据库比较,构件仓库体现了多方面的优越性,统计结果如表1所示。实践表明,本文提出的构件仓库,提高了构件的复用率,提高了企业软件开发的效率,降低了开发成本。

表1 传统的构件数据库与构件仓库对比

	传统构件数据库	构件仓库
构件复用率	32%	68%
构件查询效率	无刻面查询、复合查询,效率低,约为30%	具有刻面查询、复合查询等多种查询方式,效率高,约为90%
构件管理	无构件管理功能	具有构件管理功能,有效管理构件

4 结语

论文研究的构件仓库为构件的上传发布、查询复用、构件管理提供了一个可靠的平台,能够有效地提高软件的复用,加快软件开发的速度。构件仓库是构件应用化平台的一个基础设施,通过在企业中的应用,构件仓库的功能得到完善,同时对企业构件的管理做出贡献,使构件的使用更加方便,有效地提高了构件的复用率,提高了企业软件开发的效率,降低开发成本。本文提出的构件仓库模型还存在着许多的不足之处,希望能够在以后的应用中进一步完善。

参考文献

- 1 Mili H, Mili F, Mili A. Reusing Software: Issues and Research Directions. IEEE Trans. on Software Engineering, 1995,21(6):528-562.
- 2 杨芙清,梅宏.构件化软件设计与实现.北京:清华大学出版社,2008.100-102.
- 3 许帧.基于构件的软件开发方法及实现.软件导刊,2009,(11):17-19.
- 4 青鸟构件库概念模型.<http://sei.pku.edu.cn/jadebird/model.html>.
- 5 张清立.Maven的基本原理和Maven2的新特性.<http://blog.csdn.net/zql2002/archive/2008/06/12/2537502.aspx>. 2008.